

**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC GIA ĐỊNH  
KHOA: CÔNG NGHỆ THÔNG TIN**



# **TIỂU LUẬN**

## **QUẢN LÝ DANH SÁCH CÁC LOẠI SÁCH**

**MÔN: Thiết kế và xây dựng phần mềm**

Ngành: **CÔNG NGHỆ THÔNG TIN**

Chuyên ngành : **KỸ THUẬT PHẦN MỀM**

Giảng viên hướng dẫn: **LÊ HUỲNH PHƯỚC**

Sinh viên thực hiện: **PHAN MINH QUANG**

MSSV: **22150273**

Lớp học phần: **221403**

TP. Hồ Chí Minh, tháng 11 năm 2024

**Khoa/Viện: CÔNG NGHỆ THÔNG TIN**

**NHẬN XÉT VÀ CHẤM ĐIỂM CỦA GIẢNG VIÊN**  
**TIỂU LUẬN MÔN: THIẾT KẾ VÀ XÂY DỰNG PHẦN MỀM**

- 1. Họ và tên sinh viên: Phan Minh Quang – 22150273**
- 2. Tên đề tài: Quản lý danh sách các loại sách**
- 3. Nhận xét:**

**a) Những kết quả đạt được:**

.....

.....

.....

.....

.....

**b) Những hạn chế:**

.....

.....

.....

.....

.....

- 4. Điểm đánh giá (theo thang điểm 10, làm tròn đến 0.5):**

Sinh viên: Phan Minh Quang – 22150273

Điểm số: .....

Điểm chữ: .....

TP. HCM, ngày 18 tháng 11  
năm 2024

**Giảng viên chấm thi**  
(Ký và ghi rõ họ tên)

**NHIỆM VỤ CỦA TỪNG THÀNH VIÊN**

<b>STT</b>	<b>Họ Và Tên</b>	<b>Phân Công</b>
<b>1</b>	<b>Phan Minh Quang (Nhóm trưởng)</b>	<b>Code Chính, phân công nhiệm vụ, vẽ sơ đồ</b>
<b>2</b>	<b>Phạm Đăng Quang</b>	<b>Lập trình chức năng in danh sách sách, tìm kiếm và tính tổng sách,vẽ sơ đồ</b>
<b>3</b>	<b>Nguyễn Quang Huy</b>	<b>Lập trình chức tính trung bình sách tham khảo, xuất sách, vẽ sơ đồ</b>

## MỤC LỤC

<b>LỜI MỞ ĐẦU .....</b>	<b>1</b>
<b>CHƯƠNG 1. THU THẬP YÊU CẦU .....</b>	<b>4</b>
<b>1. Sơ đồ Use – case tổng quan.....</b>	<b>4</b>
<b>2. Mô tả Use – case.....</b>	<b>5</b>
<b>2.1. Mô tả Use – case “Add book” .....</b>	<b>5</b>
<b>2.2. Mô tả Use – case “Search book” .....</b>	<b>5</b>
<b>2.3. Mô tả Use – case chức năng “Edit book” .....</b>	<b>6</b>
<b>2.4. Mô tả Use – case “Print book” .....</b>	<b>6</b>
<b>2.5. Mô tả Use – case “Total book” .....</b>	<b>7</b>
<b>2.6. Mô tả Use – case “Delete book” .....</b>	<b>8</b>
<b>2.7. Mô tả Use – case “Get all book”.....</b>	<b>8</b>
<b>2.8 Mô tả Use – case “Average price”.....</b>	<b>9</b>
<b>2.9 Mô tả Use – case “Export book” .....</b>	<b>9</b>
<b>3. Sơ đồ trạng thái .....</b>	<b>10</b>
<b>4. Sơ đồ hoạt động .....</b>	<b>10</b>
<b>4.1. Sơ đồ hoạt động “Tính trung bình”.....</b>	<b>10</b>
<b>4.2. Sơ đồ hoạt động “Xuất” .....</b>	<b>11</b>
<b>4.3. Sơ đồ hoạt động “Thêm” .....</b>	<b>11</b>
<b>4.4. Sơ đồ hoạt động “Sửa” .....</b>	<b>12</b>
<b>4.5. Sơ đồ hoạt động “Xóa” .....</b>	<b>12</b>
<b>4.6. Sơ đồ hoạt động “Tìm kiếm” .....</b>	<b>13</b>
<b>4.7 Sơ đồ hoạt động “in” .....</b>	<b>13</b>

4.8 Sơ đồ hoạt động “Tính tổng” .....	14
<b>CHƯƠNG 2. PHÂN TÍCH YÊU CẦU .....</b>	<b>14</b>
1. Sơ đồ tuần tự mức phân tích .....	14
1.1 Sơ đồ tuần tự “AddBook” .....	14
1.2 Sơ đồ tuần tự “EditBook” .....	16
1.3 Sơ đồ tuần tự “SearchBook” .....	18
1.4 Sơ đồ tuần tự “DeleteBook” .....	21
1.5 Sơ đồ tuần tự “PrintBook” .....	23
1.6 Sơ đồ tuần tự “TotalBook” .....	25
1.7 Sơ đồ tuần tự “GetBookList” .....	27
1.8 Sơ đồ tuần tự “AverageUnitPrice” .....	28
1.9 Sơ đồ tuần tự “ExportBook” .....	30
2. Sơ đồ class mức phân tích.....	31
<b>CHƯƠNG 3. KIỂM THỬ.....</b>	<b>32</b>
1. Test case chức năng “addbook” .....	32
2. Test case chức năng “editbook” .....	34
3. Test case chức năng “deletebook” .....	36
4. Test case chức năng “printbook” .....	39
5. Test case chức năng “searchbook” .....	41
6. Test case chức năng “totalbookprice” .....	45
7. Test case chức năng “averageunitprice” .....	49
<b>CHƯƠNG 4: KẾT QUẢ .....</b>	<b>55</b>
1. Đánh giá phần mềm.....	55
2. Hình ảnh tất cả các chức năng của chương trình.....	56
2.1. Chức năng lấy thông tin tất cả sách.....	56

2.2. Chức năng thêm sách .....	56
2.3. Chức năng sửa sách .....	57
2.4. Chức năng xóa sách .....	57
2.5. Chức năng tìm kiếm .....	57
2.6. Chức năng in .....	58
2.7. Chức năng tính tổng .....	58
2.8. Chức năng tính trung bình cộng .....	59
2.9. Chức năng xuất sách theo nhà xuất bản .....	59
HƯỚNG PHÁT TRIỂN .....	60
<u><a href="#">TÀI LIỆU THAM KHẢO</a></u> .....	61

## **LỜI MỞ ĐẦU**

### **1. Tên đề tài**

Quản lý danh sách các loại sách

### **2. Mục tiêu của đề tài**

"Quản lý Danh Sách Sách trong Thư viện" là xây dựng một hệ thống phần mềm giúp quản lý và theo dõi các sách trong thư viện một cách hiệu quả và dễ dàng. Hệ thống sẽ cho phép người dùng thêm mới, xóa, cập nhật và tìm kiếm sách trong cơ sở dữ liệu, đồng thời hiển thị thông tin chi tiết về sách như tên sách, tác giả, giá trị và tình trạng tồn kho. Giao diện người dùng sẽ được thiết kế thân thiện và dễ sử dụng, tối ưu hóa cho các chức năng cần thiết như tra cứu sách, thống kê giá trị sách và tính toán giá trị trung bình của các sách tham khảo. Một trong những mục tiêu chính của hệ thống là đảm bảo dữ liệu được quản lý một cách chính xác, nhanh chóng, và có thể hỗ trợ nhiều chức năng khác nhau.

### **3. Phạm vi của đề tài**

Đề tài "Quản lý Danh Sách Sách trong Thư viện" bao gồm các nội dung và chức năng chính nhằm phát triển một hệ thống phần mềm quản lý sách trong thư viện. Trước hết, đề tài tập trung vào việc xây dựng và triển khai các chức năng cơ bản như thêm mới, xóa, cập nhật thông tin sách, và tìm kiếm sách trong hệ thống cơ sở dữ liệu. Hệ thống sẽ hỗ trợ việc quản lý thông tin sách như tên, tác giả, thể loại, giá trị, tình trạng tồn kho và năm xuất bản.

Hệ thống sẽ có các tính năng như tính toán giá trị trung bình của sách tham khảo, thống kê số lượng sách theo thể loại, in báo cáo và xuất danh sách sách ra các định dạng phổ biến như PDF hoặc CSV. Đặc biệt, việc thiết kế giao diện người dùng (UI) sẽ được chú trọng để đơn giản, dễ sử

dụng và trực quan, giúp người dùng dễ dàng thao tác trên cả thiết bị di động và máy tính để bàn.

Về mặt kỹ thuật, hệ thống sẽ tối ưu hóa hiệu suất, đảm bảo dữ liệu được lưu trữ an toàn và dễ dàng truy xuất. Đảm bảo tính tương thích với nhiều môi trường hoạt động và dễ dàng mở rộng trong tương lai để tích hợp thêm các chức năng nâng cao nếu cần.

#### **4. Ý nghĩa của đề tài**

Đề tài "Quản lý Danh Sách Sách trong Thư viện" mang lại giá trị thiết thực trong việc hỗ trợ các thư viện và tổ chức giáo dục quản lý kho sách một cách hiệu quả và chính xác. Hệ thống giúp các quản lý thư viện dễ dàng theo dõi tình trạng sách, tình hình mượn trả, và các thông tin chi tiết của từng cuốn sách, đáp ứng nhu cầu lưu trữ và xử lý thông tin trong môi trường số hóa hiện nay.

Thứ hai, hệ thống này giúp nâng cao hiệu quả công việc của nhân viên thư viện và các độc giả thông qua các tính năng tìm kiếm, cập nhật và báo cáo nhanh chóng. Với giao diện người dùng dễ sử dụng và thân thiện, người dùng sẽ có trải nghiệm mượt mà và tiết kiệm thời gian khi tìm kiếm và quản lý sách, đồng thời giúp giảm thiểu sai sót trong công tác nhập liệu và theo dõi thông tin sách.

Hơn nữa, đề tài này còn có ý nghĩa trong việc phát triển và ứng dụng các công nghệ lập trình phần mềm, đặc biệt là trong lĩnh vực quản lý dữ liệu và hệ thống thông tin. Hệ thống không chỉ đảm bảo tính ổn định và hiệu suất cao mà còn giúp nghiên cứu và phát triển các giải pháp phần mềm tiên tiến trong lĩnh vực quản lý thông tin.

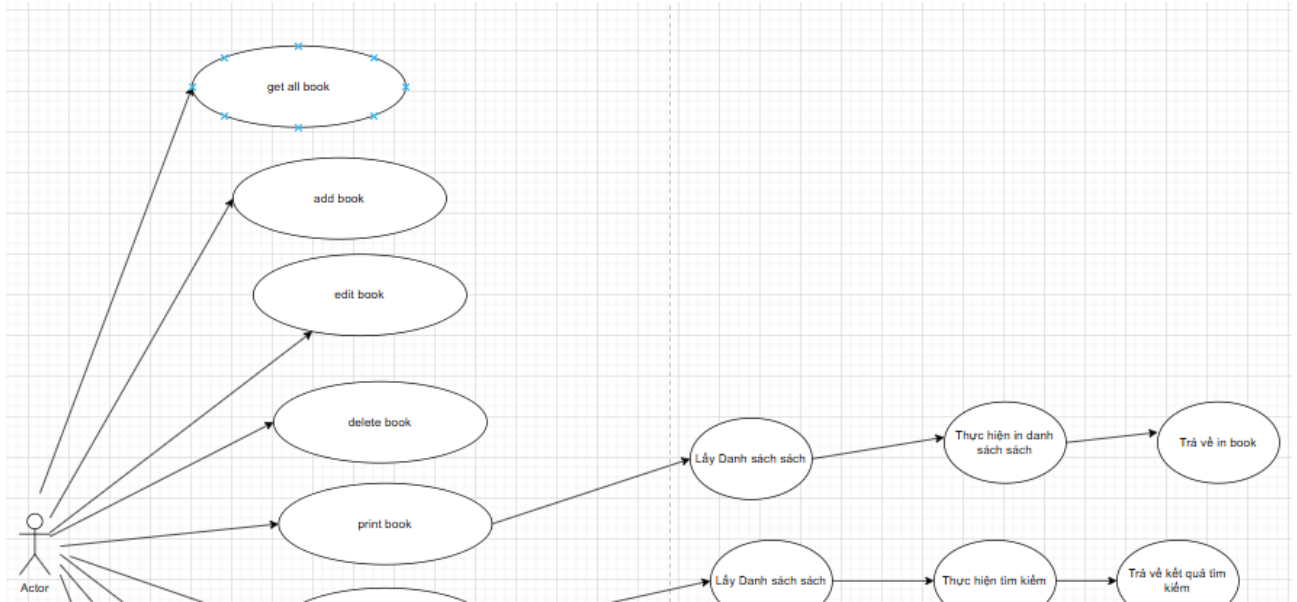
Cuối cùng, hệ thống này không chỉ phục vụ nhu cầu quản lý sách trong thư viện mà còn có thể được áp dụng trong các lĩnh vực khác như giáo dục, tổ chức sự kiện, và nghiên cứu khoa học, góp phần nâng cao năng suất làm việc và khả năng quản lý dữ liệu một cách khoa học và hiệu quả.



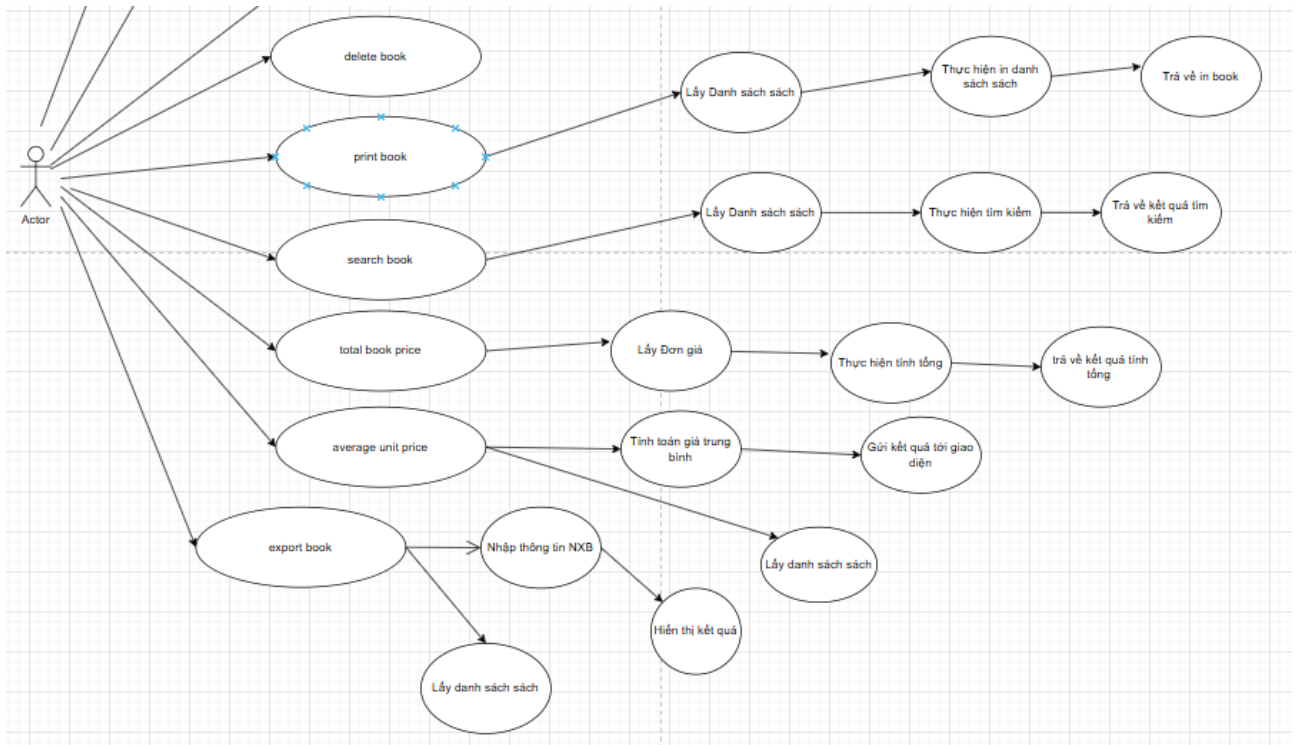


# CHƯƠNG 1. THU THẬP YÊU CẦU

## 1. Sơ đồ Use – case tổng quan



**Hình 1.1 Sơ đồ Use – case tổng quan**



**Hình 1.2 Sơ đồ Use – case tổng quan**

## 2. Mô tả Use – case

### 2.1. Mô tả Use – case “Add book”

Add book	
Input	Mã sách, ngày nhập (ngày, tháng, năm), đơn giá, số lượng, nhà xuất bản, tình trạng (mới, cũ), thuế
Output	Trả về kết quả xử lý
Primary course	
1	nhận thông tin sách
2	kiểm tra hợp lệ - ngày nhập (dd/mm/yyyy)
3	hợp lệ: lưu / thêm sách
4	không hợp lệ: cảnh báo người dùng
5	

### 2.2. Mô tả Use – case “Search book”

Search Book	
Input	nhận id book
Output	kết quả xử lý trả về thông tin book tìm kiếm lên cho người dùng
Primary course	
1	Kiểm tra id có tồn tại trong danh sách sách ko
2	nếu không tồn tại báo lỗi- ngày nhập
3	nếu tồn tại trả về dữ liệu book

4	hiển thị dữ liệu book
5	

### 2.3. Mô tả Use – case chức năng “Edit book”

Edit book	
Input	nhận id book
Output	trả về kết quả xử lý
Primary course	
1	nhận id book cần sửa
2	kiểm tra id sách tồn tại không
3	sửa thông tin
4	kiểm tra hợp lệ
5	hợp lệ :cập nhật thông tin sách
6	không hợp lệ: cảnh báo người dùng

### 2.4. Mô tả Use – case “Print book”

Print Book	
Input	
Output	kết quả xử lý trả về thông tin in danh sách sách lên cho người dùng
Primary course	

1	Nhận thông tin id sách
2	Kiểm tra id có tồn tại trong database hay ko
3	nếu tồn tại trả về thông tin sách
4	nếu không tồn tại thì báo lỗi
5	

### 2.5. Mô tả Use – case “Total book”

Total Book	
Input	
Output	kết quả xử lý trả về kết quả xử lý
Primary course	
1	nhận đơn giá
2	Kiểm tra có tồn tại trong database hay ko
3	<p>nếu tồn tại : Thực hiện tính tổng các loại sách</p> <p>-Nếu sách mới: Thành tiền = Số Lượng * Đơn Giá</p> <p>-Nếu sách cũ: Thành tiền = Số Lượng * Đơn Giá * 50%</p> <p>-Sách Tham Khảo : Thành Tiền = Số Lượng * Đơn Giá + Thuế</p> <p>- Tính Cộng dồn các loại sách</p>
4	nếu ko tồn tại báo lỗi
5	trả về kết quả

### 2.6. Mô tả Use – case “Delete book”

Delete Book	
Input	nhận id book
Output	trả về kết quả xử lý
Primary course	
1	Nhận id sách cần xóa
2	Không tìm thấy sách
3	báo lỗi sách không tồn tại
4	Tìm thấy sách cần xóa
5	Xóa sách ra khỏi danh sách
6	trả về kết quả xử lý

### 2.7. Mô tả Use – case “Get all book”

Get all book	
Input	
Output	trả về danh sách book
Primary course	
1	thực hiện lấy sách
2	trả về danh sách sách
3	
4	
5	

6	
---	--

### 2.8 Mô tả Use – case “Average price”

Average price	
Input	
Output	trả về kết quả xử lý
Primary course	
1	nhận đơn giá
2	Kiểm tra có tồn tại trong database hay ko
3	nếu tồn tại : Thực hiện tính trung bình các loại sách
4	nếu ko tồn tại báo lỗi
5	trả về kết quả
6	

### 2.9 Mô tả Use – case “Export book”

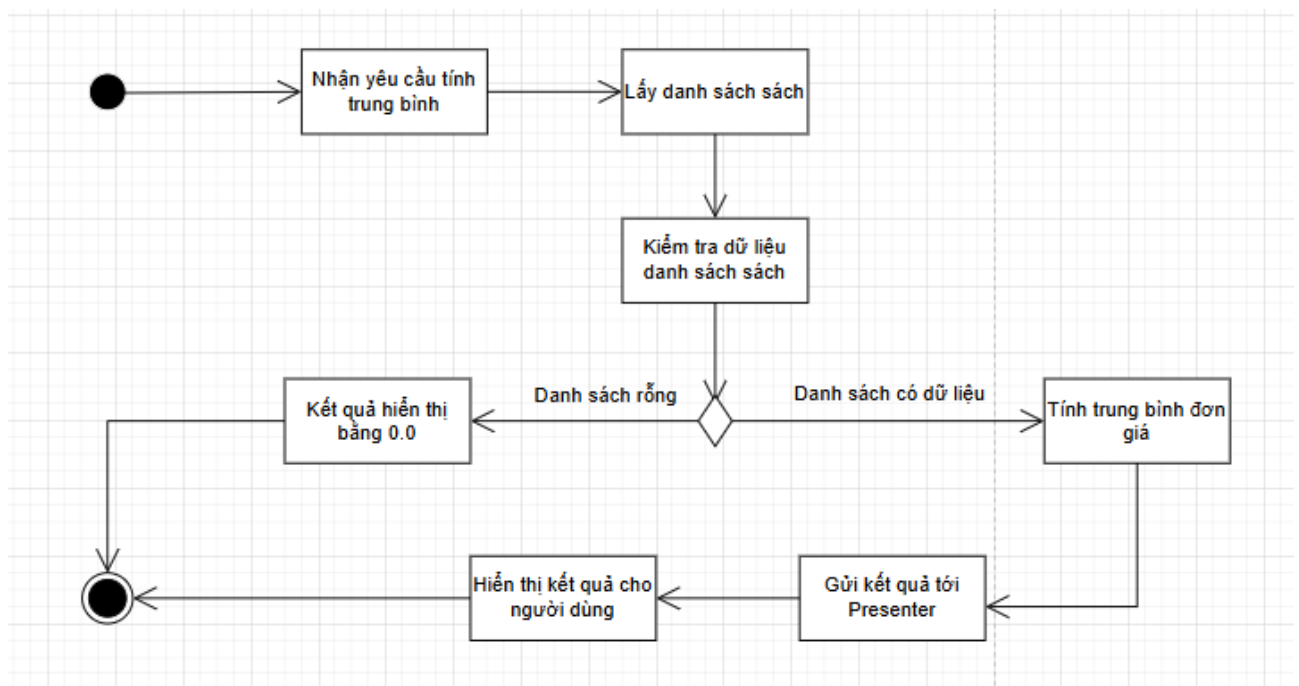
Export book	
Input	
Output	kết quả xử lý trả về thông tin in danh sách lên cho người dùng
Primary course	
1	Nhận thông tin publisher

2	Kiểm tra có tồn tại trong database hay ko
3	nếu tồn tại trả về thông tin publisher
4	nếu không tồn tại thì báo lỗi
5	trả về kết quả
6	

### 3. Sơ đồ trạng thái

### 4. Sơ đồ hoạt động

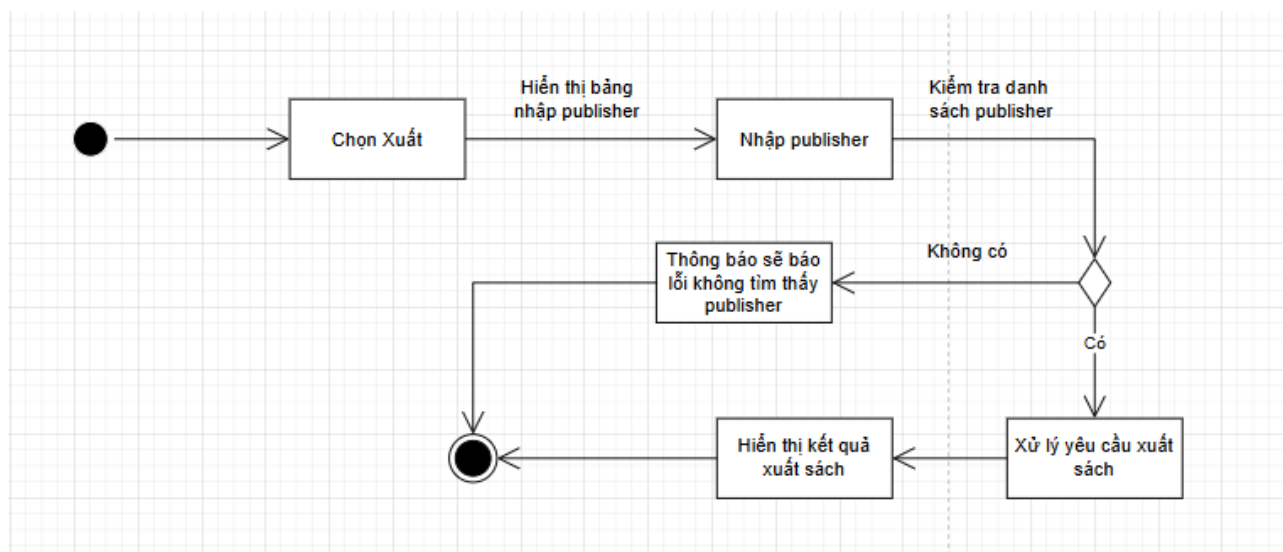
#### 4.1. Sơ đồ hoạt động “Tính trung bình”



**Hình 1.3 Sơ đồ hoạt động tính trung bình**

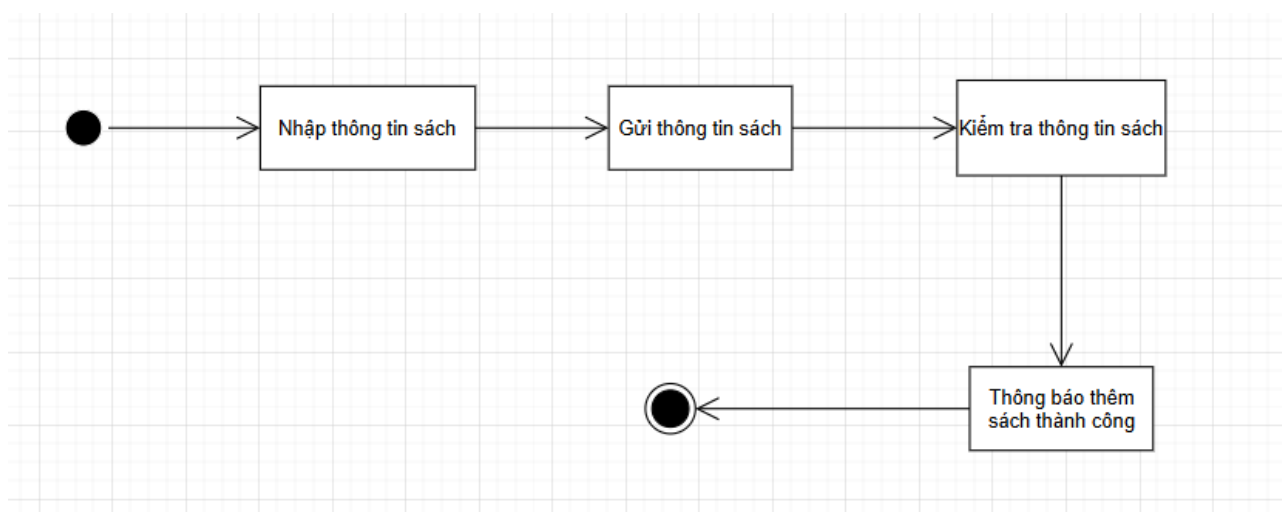


#### 4.2. Sơ đồ hoạt động “Xuất”



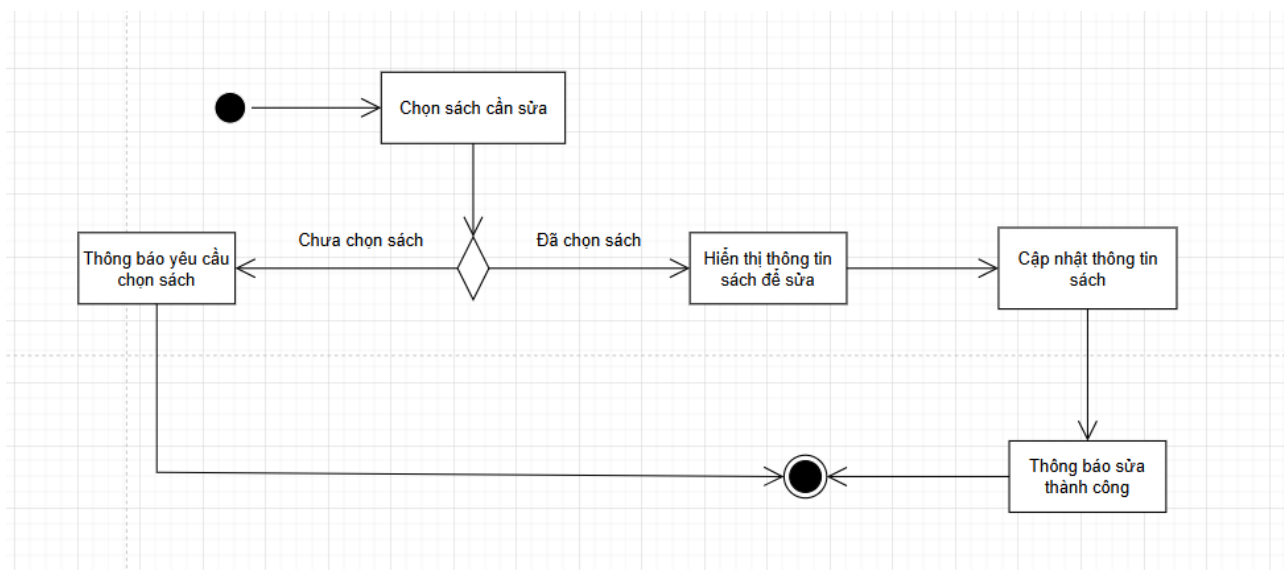
Hình 1.4 Sơ đồ hoạt động xuất

#### 4.3. Sơ đồ hoạt động “Thêm”



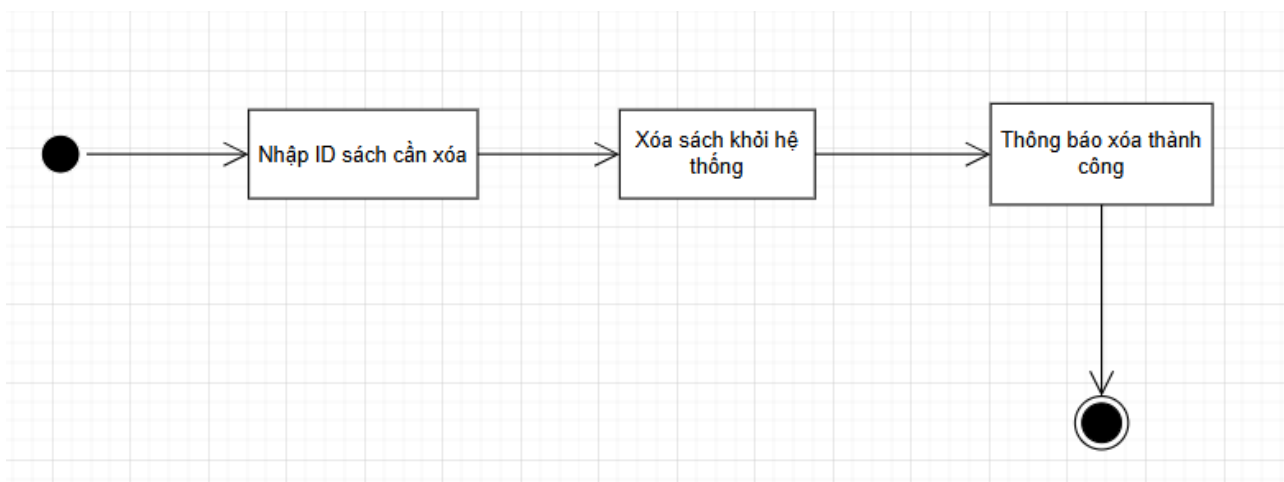
Hình 1.5 Sơ đồ hoạt động thêm

#### 4.4. Sơ đồ hoạt động “Sửa”



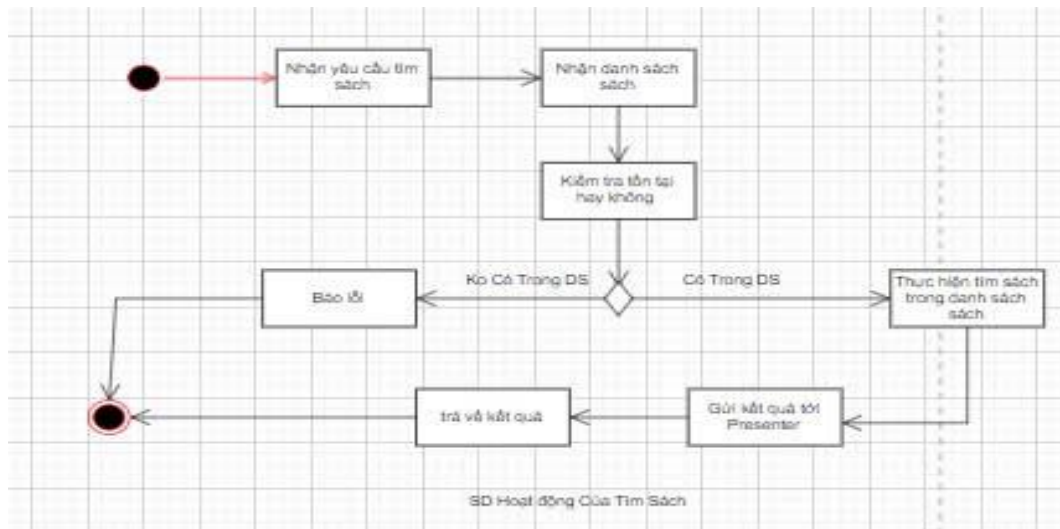
Hình 1.6 Sơ đồ hoạt động thêm

#### 4.5. Sơ đồ hoạt động “Xóa”



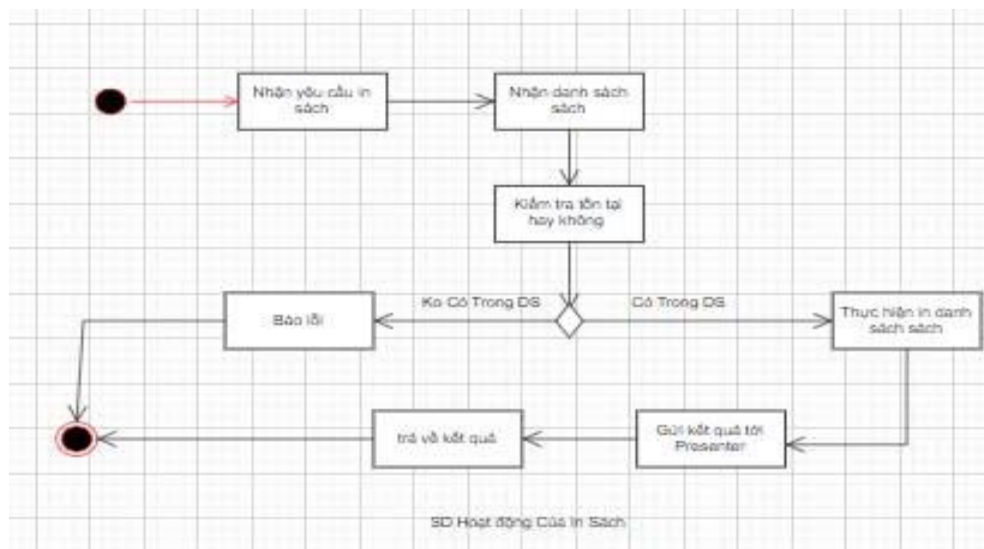
Hình 1.7 Sơ đồ hoạt động xóa

#### 4.6. Sơ đồ hoạt động “Tìm kiếm”



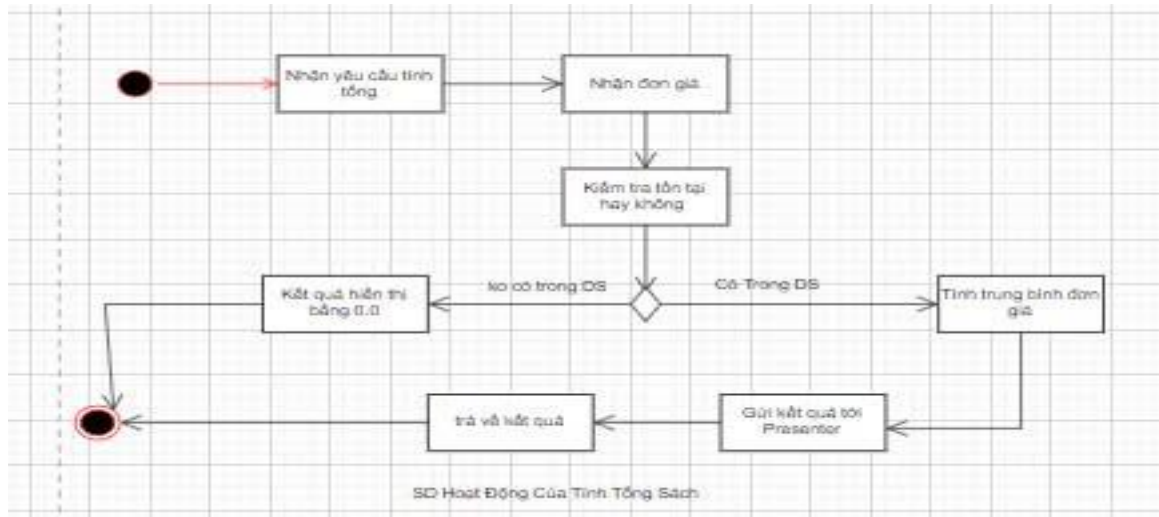
Hình 1.8 Sơ đồ hoạt động tìm kiếm

#### 4.7 Sơ đồ hoạt động “in”



Hình 1.9 Sơ đồ hoạt động in

## 4.8 Sơ đồ hoạt động “Tính tổng”

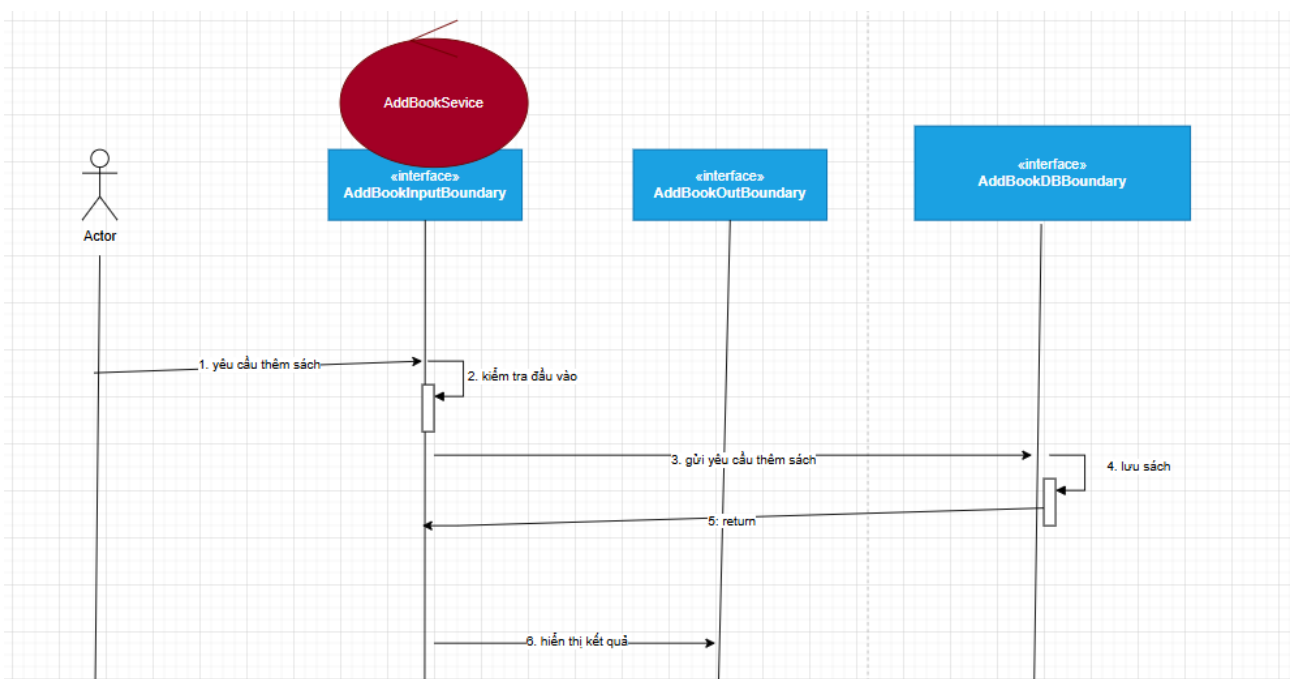


Hình 1.10 Sơ đồ hoạt động tính tổng

## CHƯƠNG 2. PHÂN TÍCH YÊU CẦU

### 1. Sơ đồ tuần tự mức phân tích

#### 1.1 Sơ đồ tuần tự “AddBook”



## Thành phần chính

- **Actor:** Đây là người sử dụng hệ thống, có thể là quản trị viên hoặc nhân viên nhập liệu, thực hiện thao tác thêm sách.
- **AddBookService:** Là lớp điều khiển trung tâm, chịu trách nhiệm xử lý logic chính của quy trình thêm sách. Lớp này đóng vai trò điều phối giữa các thành phần Input Boundary, Output Boundary, và Database Boundary.
- **AddBookInputBoundary:** Là giao diện nhận yêu cầu từ người dùng, chịu trách nhiệm chuyển yêu cầu sang lớp điều khiển chính.
- **AddBookOutBoundary:** Là giao diện gửi thông tin từ lớp điều khiển đến thành phần lưu trữ dữ liệu (Database Boundary).
- **AddBookDBBoundary:** Là giao diện kết nối với cơ sở dữ liệu, thực hiện thao tác lưu sách vào hệ thống.

## Luồng hoạt động

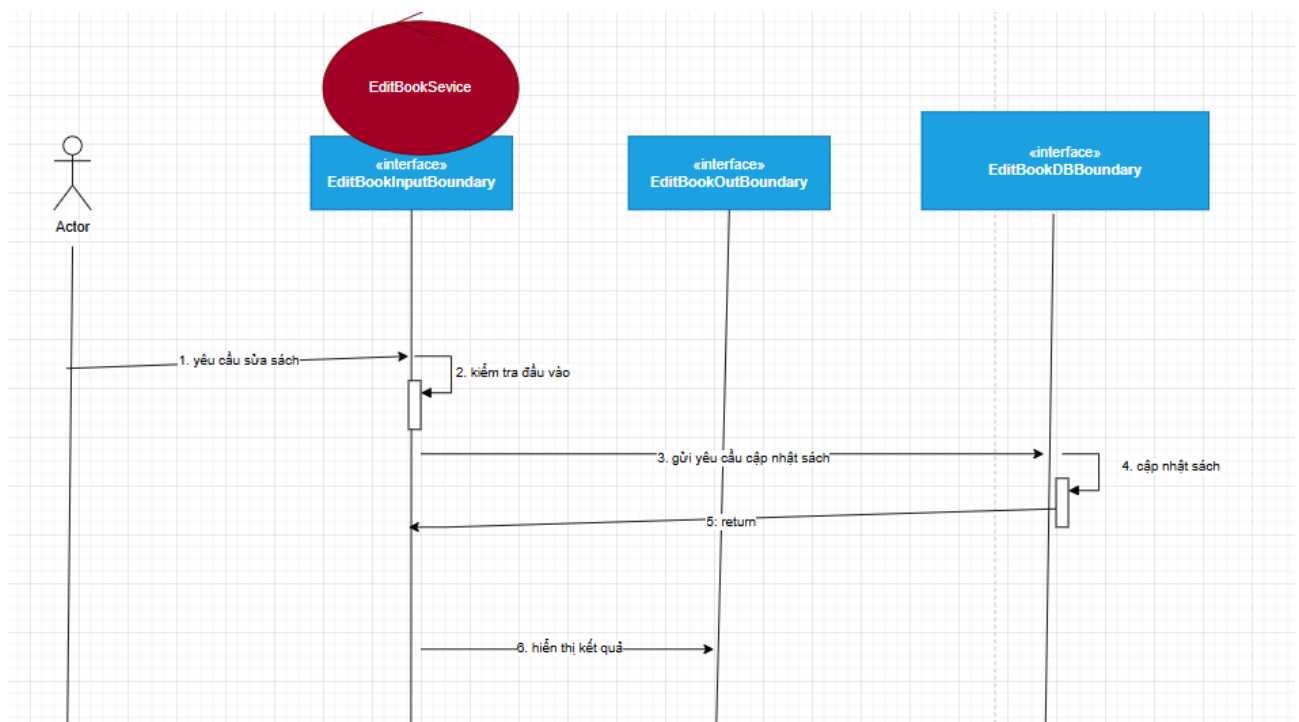
- **Actor thực hiện yêu cầu:** Người dùng gửi yêu cầu thêm sách qua phương thức execute(RequestBookData) tới giao diện AddBookInputBoundary.
- **Xử lý logic chính:** Giao diện AddBookInputBoundary chuyển yêu cầu tới lớp điều khiển AddBookService. Lớp điều khiển xử lý logic và kiểm tra tính hợp lệ của dữ liệu đầu vào.
- **Gửi yêu cầu tới Database Boundary:** Sau khi xử lý xong, lớp điều khiển chuyển dữ liệu qua giao diện AddBookOutBoundary, giao diện này gửi lệnh insert(RequestBookData) tới thành phần AddBookDBBoundary.
- **Thao tác cơ sở dữ liệu:** AddBookDBBoundary thực hiện thao tác lưu dữ liệu sách vào cơ sở dữ liệu.
- **Trả kết quả:** Sau khi thêm dữ liệu thành công, cơ sở dữ liệu trả kết quả lại cho lớp điều khiển thông qua các giao diện.

- **Hiển thị kết quả:** Lớp điều khiển trả về kết quả qua phương thức `showResult(ResponseResult)` để hiển thị cho người dùng.

### Ý nghĩa của các bước:

- Step 1: Người dùng gửi yêu cầu thêm sinh viên, khởi đầu luồng xử lý.
- Step 2: Lớp điều khiển kiểm tra dữ liệu đầu vào, đảm bảo tính hợp lệ trước khi thực hiện lưu trữ.
- Step 3-4: Lớp điều khiển giao tiếp với giao diện cơ sở dữ liệu, đảm bảo tách biệt trách nhiệm giữa xử lý logic và lưu trữ dữ liệu.
- Step 5-6: Hệ thống phản hồi kết quả cho người dùng, đảm bảo tính minh bạch và dễ dàng theo dõi.

### 1.2 Sơ đồ tuần tự “EditBook”



### Thành phần chính:

- **Actor:** Đây là người sử dụng hệ thống, có thể là quản trị viên hoặc nhân viên nhập liệu, thực hiện thao tác chỉnh sửa thông tin sách.
- **EditBookService:** Lớp điều khiển trung tâm, chịu trách nhiệm xử lý logic chính của quy trình chỉnh sửa sách. Lớp này đóng vai trò điều phối giữa các thành phần Input Boundary, Output Boundary, và Database Boundary.
- **EditBookInputBoundary:** Là giao diện nhận yêu cầu từ người dùng, chịu trách nhiệm chuyển yêu cầu chỉnh sửa sách sang lớp điều khiển chính.
- **EditBookOutBoundary:** Là giao diện gửi thông tin từ lớp điều khiển đến thành phần lưu trữ dữ liệu (Database Boundary) và phản hồi kết quả lại cho người dùng.
- **EditBookDBBoundary:** Là giao diện kết nối với cơ sở dữ liệu, thực hiện thao tác cập nhật thông tin sách trong hệ thống.

#### **Luồng hoạt động:**

- Actor thực hiện yêu cầu: Người dùng gửi yêu cầu chỉnh sửa sách qua phương thức execute(RequestBookUpdateData) tới giao diện EditBookInputBoundary.
- Xử lý logic chính:
  - Giao diện EditBookInputBoundary chuyển yêu cầu tới lớp điều khiển EditBookService.
  - EditBookService xử lý logic và kiểm tra dữ liệu đầu vào, đảm bảo rằng thông tin cập nhật của sách hợp lệ (ví dụ: kiểm tra xem mã sách tồn tại hay thông tin cập nhật có đầy đủ không).
  - Nếu dữ liệu không hợp lệ, lớp điều khiển trả về lỗi cho người dùng qua giao diện EditBookOutBoundary.
  - Nếu dữ liệu hợp lệ, lớp điều khiển tiếp tục xử lý và chuẩn bị dữ liệu để gửi tới cơ sở dữ liệu.

- Gửi yêu cầu tới Database Boundary:
  - Sau khi xử lý xong, lớp điều khiển chuyển dữ liệu qua giao diện EditBookOutBoundary.
  - Giao diện này gửi lệnh update(RequestBookUpdateData) tới thành phần EditBookDBBoundary để thực hiện thao tác cập nhật thông tin sách.
- Thao tác cơ sở dữ liệu:
  - EditBookDBBoundary thực hiện thao tác cập nhật thông tin sách trong cơ sở dữ liệu.
  - Nếu xảy ra lỗi (ví dụ: không tìm thấy mã sách), hệ thống sẽ trả về thông báo lỗi qua giao diện EditBookOutBoundary.
  - Nếu thành công, cơ sở dữ liệu gửi xác nhận lại lớp điều khiển.
- Trả kết quả:
  - Sau khi nhận được phản hồi từ cơ sở dữ liệu, lớp điều khiển thông báo kết quả xử lý qua giao diện EditBookOutBoundary.

### *1.3 Sơ đồ tuần tự “SearchBook”*

#### **Thành phần chính:**

- **Actor:** Người thực hiện thao tác tìm kiếm sách, có thể là quản trị viên, nhân viên, hoặc người dùng cuối của hệ thống.
- **SearchBookService:** Lớp điều khiển trung tâm, chịu trách nhiệm xử lý logic chính của quy trình tìm kiếm sách. Lớp này điều phối giữa các thành phần Input Boundary, Output Boundary, và Database Boundary.
- **SearchBookInputBoundary:** Giao diện nhận yêu cầu tìm kiếm từ người dùng và chuyển thông tin tới lớp điều khiển chính.



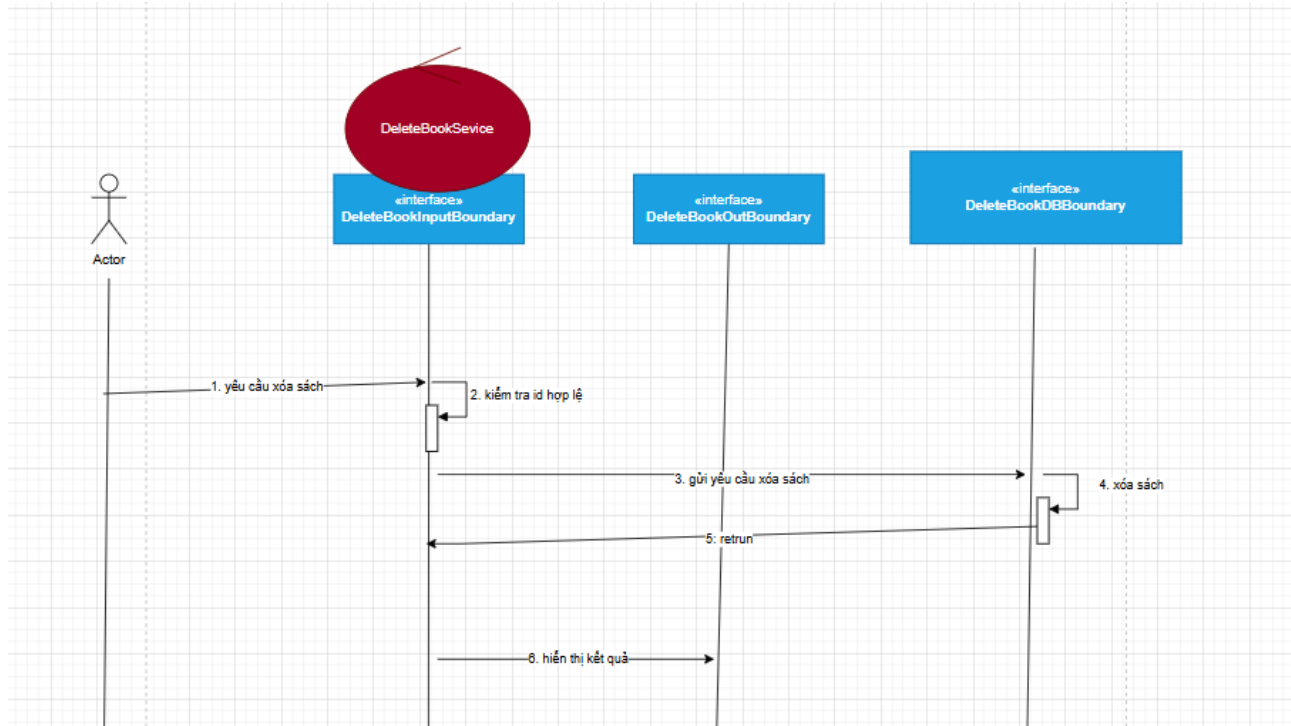
- **SearchBookOutBoundary:** Giao diện gửi yêu cầu từ lớp điều khiển đến cơ sở dữ liệu để tìm kiếm sách và phản hồi kết quả tìm kiếm lại cho người dùng.
- **SearchBookDatabaseBoundary:** Giao diện kết nối với cơ sở dữ liệu, thực hiện thao tác truy xuất thông tin sách từ hệ thống.

#### **Luồng hoạt động:**

- Actor thực hiện yêu cầu: Người dùng gửi yêu cầu tìm kiếm sách thông qua phương thức search(RequestSearchCriteria) tới giao diện SearchBookInputBoundary.
- Xử lý logic chính:
  - Giao diện SearchBookInputBoundary chuyển yêu cầu tới lớp điều khiển SearchBookService.
  - SearchBookService xử lý logic, kiểm tra tiêu chí tìm kiếm, và đảm bảo rằng yêu cầu hợp lệ (ví dụ: kiểm tra thông tin như mã sách, tên sách, hoặc tác giả không bị thiếu).
  - Nếu dữ liệu đầu vào không hợp lệ, lớp điều khiển trả về lỗi qua giao diện SearchBookOutBoundary để hiển thị thông báo cho người dùng.
  - Nếu hợp lệ, lớp điều khiển tiếp tục gửi yêu cầu tìm kiếm tới cơ sở dữ liệu.
- Gửi yêu cầu tới Database Boundary:
  - Lớp điều khiển chuyển yêu cầu tìm kiếm qua giao diện SearchBookOutBoundary.
  - Giao diện này gửi lệnh searchBooks(RequestSearchCriteria) tới thành phần SearchBookDatabaseBoundary.
- Thao tác cơ sở dữ liệu:

- SearchBookDatabaseBoundary thực hiện thao tác truy vấn thông tin sách trong cơ sở dữ liệu dựa trên tiêu chí tìm kiếm.
- Nếu không tìm thấy sách phù hợp, cơ sở dữ liệu trả về kết quả rỗng hoặc thông báo lỗi qua giao diện SearchBookOutBoundary.
- Nếu tìm thấy, cơ sở dữ liệu trả về danh sách các sách phù hợp qua giao diện này.
- Trả kết quả:
  - Lớp điều khiển nhận danh sách kết quả từ cơ sở dữ liệu và định dạng lại thông tin (nếu cần).
  - Lớp điều khiển trả về kết quả qua giao diện SearchBookOutBoundary để hiển thị cho người dùng.
- Hiển thị kết quả:
  - Giao diện gửi kết quả thông qua phương thức showResults(ResponseSearchResults) để hiển thị danh sách các sách phù hợp với yêu cầu tìm kiếm của người dùng.

### 1.4 Sơ đồ tuần tự “DeleteBook”



#### Thành phần chính

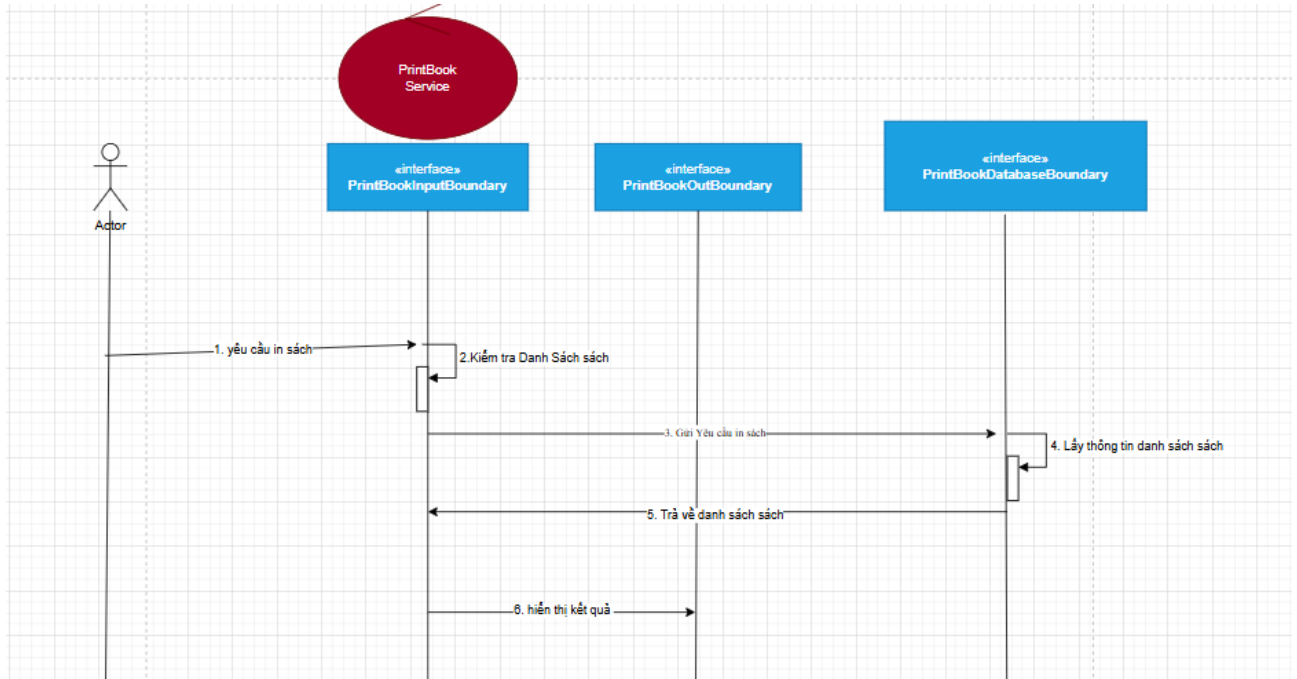
- **Actor:** Người thực hiện thao tác xóa sách, có thể là quản trị viên hoặc nhân viên quản lý thư viện.
- **DeleteBookService:** Lớp điều khiển trung tâm, chịu trách nhiệm xử lý logic chính của quy trình xóa sách. Lớp này điều phối giữa các thành phần Input Boundary, Output Boundary, và Database Boundary.
- **DeleteBookInputBoundary:** Giao diện nhận yêu cầu xóa sách từ người dùng và chuyển thông tin tới lớp điều khiển chính.
- **DeleteBookOutBoundary:** Giao diện gửi yêu cầu từ lớp điều khiển đến cơ sở dữ liệu để xóa sách và phản hồi kết quả lại cho người dùng.
- **DeleteBookDBBoundary:** Giao diện kết nối với cơ sở dữ liệu, thực hiện thao tác xóa sách khỏi hệ thống.

#### Luồng hoạt động

- Actor gửi yêu cầu xóa sách thông qua phương thức `delete(RequestBookId)` tới giao diện `DeleteBookInputBoundary`.

- Giao diện DeleteBookInputBoundary chuyển tiếp yêu cầu tới lớp điều khiển DeleteBookService.
- DeleteBookService xử lý logic, kiểm tra tính hợp lệ của ID sách (ví dụ: xác minh ID sách có tồn tại hay không).
  - Nếu ID không hợp lệ, lớp điều khiển trả về lỗi qua giao diện DeleteBookOutBoundary để hiển thị thông báo lỗi cho người dùng.
  - Nếu hợp lệ, yêu cầu xoá sách sẽ được gửi tới cơ sở dữ liệu.
- Lớp điều khiển chuyển yêu cầu xoá sách qua giao diện DeleteBookOutBoundary.
- Giao diện này gửi lệnh deleteBook(RequestBookId) tới thành phần DeleteBookDBBoundary.
- DeleteBookDBBoundary thực hiện thao tác xoá sách trong cơ sở dữ liệu:
  - Nếu không thể xoá (ví dụ: sách đang được mượn hoặc không tìm thấy ID), cơ sở dữ liệu trả về lỗi qua giao diện DeleteBookOutBoundary.
  - Nếu xoá thành công, cơ sở dữ liệu trả về xác nhận đã xoá sách.
- Lớp điều khiển nhận kết quả từ cơ sở dữ liệu, định dạng lại thông tin (nếu cần), và chuyển dữ liệu qua giao diện DeleteBookOutBoundary để gửi thông báo đến người dùng.
- Giao diện hiển thị kết quả thông qua phương thức showResult(ResponseResult), thông báo rằng sách đã được xoá thành công hoặc không thể xoá.

### 1.5 Sơ đồ tuần tự “PrintBook”



#### Thành phần chính

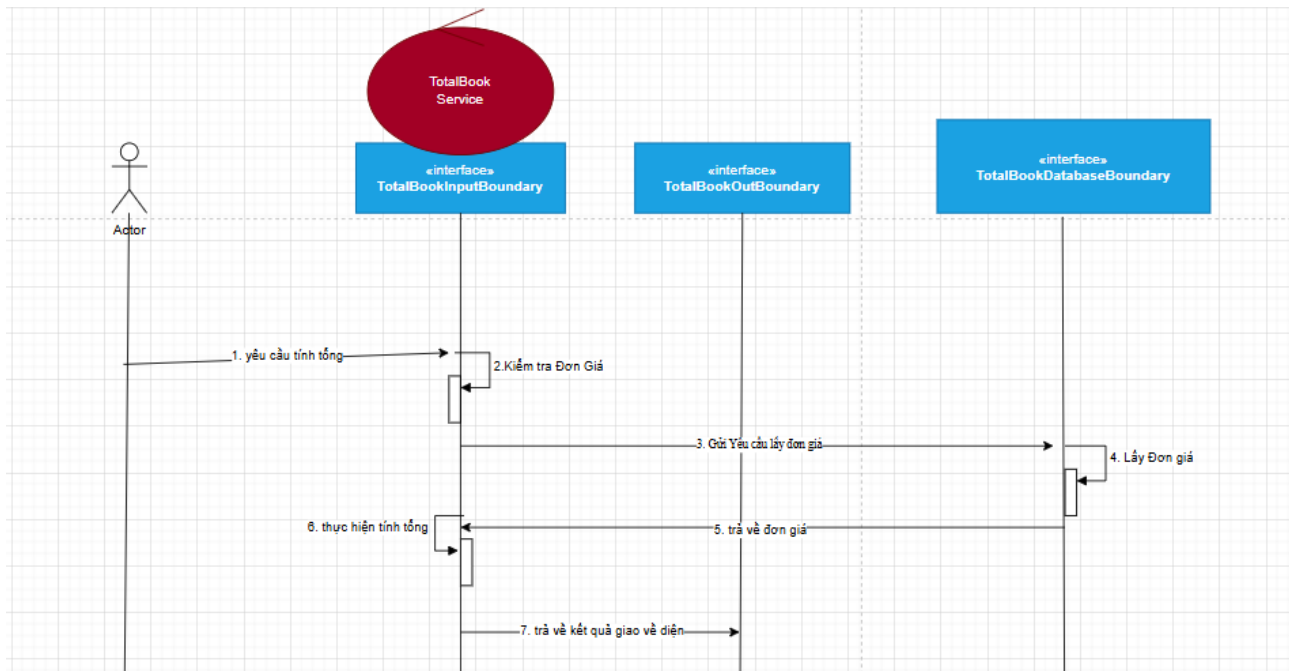
- **Actor:** Người yêu cầu in danh sách sách, thường là quản trị viên hoặc nhân viên thư viện.
- **PrintBookService:** Lớp điều khiển trung tâm, chịu trách nhiệm xử lý yêu cầu in danh sách sách. Lớp này điều phối giữa các thành phần Input Boundary, Output Boundary, và Database Boundary.
- **PrintBookInputBoundary:** Giao diện tiếp nhận yêu cầu in danh sách sách từ người dùng và chuyển thông tin đến lớp điều khiển chính.
- **PrintBookOutBoundary:** Giao diện đảm nhận việc gửi yêu cầu lấy danh sách sách từ lớp điều khiển đến cơ sở dữ liệu và phản hồi kết quả về cho người dùng.
- **PrintBookDatabaseBoundary:** Giao diện kết nối trực tiếp với cơ sở dữ liệu, thực hiện thao tác truy vấn thông tin danh sách sách từ hệ thống.

#### Luồng hoạt động

- Actor gửi yêu cầu in danh sách sách thông qua giao diện PrintBookInputBoundary.

- Giao diện `PrintBookInputBoundary` chuyển tiếp yêu cầu đến lớp điều khiển `PrintBookService`.
- `PrintBookService` kiểm tra yêu cầu, xác minh tính hợp lệ của thông tin đầu vào (ví dụ: danh sách sách có tồn tại không).
  - Nếu thông tin không hợp lệ, lớp điều khiển trả về thông báo lỗi qua giao diện `PrintBookOutBoundary` để hiển thị cho người dùng.
  - Nếu thông tin hợp lệ, yêu cầu được chuyển tiếp đến cơ sở dữ liệu.
- Lớp điều khiển gửi yêu cầu lấy danh sách sách qua giao diện `PrintBookOutBoundary`.
- Giao diện này gửi lệnh truy vấn `fetchBookList()` đến thành phần `PrintBookDatabaseBoundary`.
- `PrintBookDatabaseBoundary` truy vấn cơ sở dữ liệu để lấy danh sách sách:
  - Nếu không tìm thấy dữ liệu, cơ sở dữ liệu trả về thông báo lỗi qua giao diện `PrintBookOutBoundary`.
  - Nếu truy vấn thành công, cơ sở dữ liệu trả về danh sách sách.
- Lớp điều khiển nhận dữ liệu danh sách sách từ cơ sở dữ liệu, định dạng lại thông tin (nếu cần), và chuyển thông tin qua giao diện `PrintBookOutBoundary`.
- Giao diện hiển thị kết quả qua phương thức `showResult(ResponseBookList)` để cung cấp danh sách sách cho người dùng.

## 1.6 Sơ đồ tuần tự “TotalBook”



### Thành phần chính

- **Actor:** Người yêu cầu tính tổng giá trị sách, có thể là quản trị viên hoặc nhân viên quản lý thư viện.
- **TotalBookService:** Lớp điều khiển trung tâm, chịu trách nhiệm xử lý logic chính của quy trình tính tổng giá trị sách. Lớp này điều phối giữa các thành phần Input Boundary, Output Boundary, và Database Boundary.
- **TotalBookInputBoundary:** Giao diện tiếp nhận yêu cầu tính tổng từ người dùng và chuyển thông tin đến lớp điều khiển chính.
- **TotalBookOutBoundary:** Giao diện đảm nhận việc gửi yêu cầu lấy dữ liệu từ cơ sở dữ liệu và trả kết quả tính toán cho người dùng.
- **TotalBookDatabaseBoundary:** Giao diện kết nối trực tiếp với cơ sở dữ liệu, thực hiện thao tác truy xuất thông tin đơn giá sách.

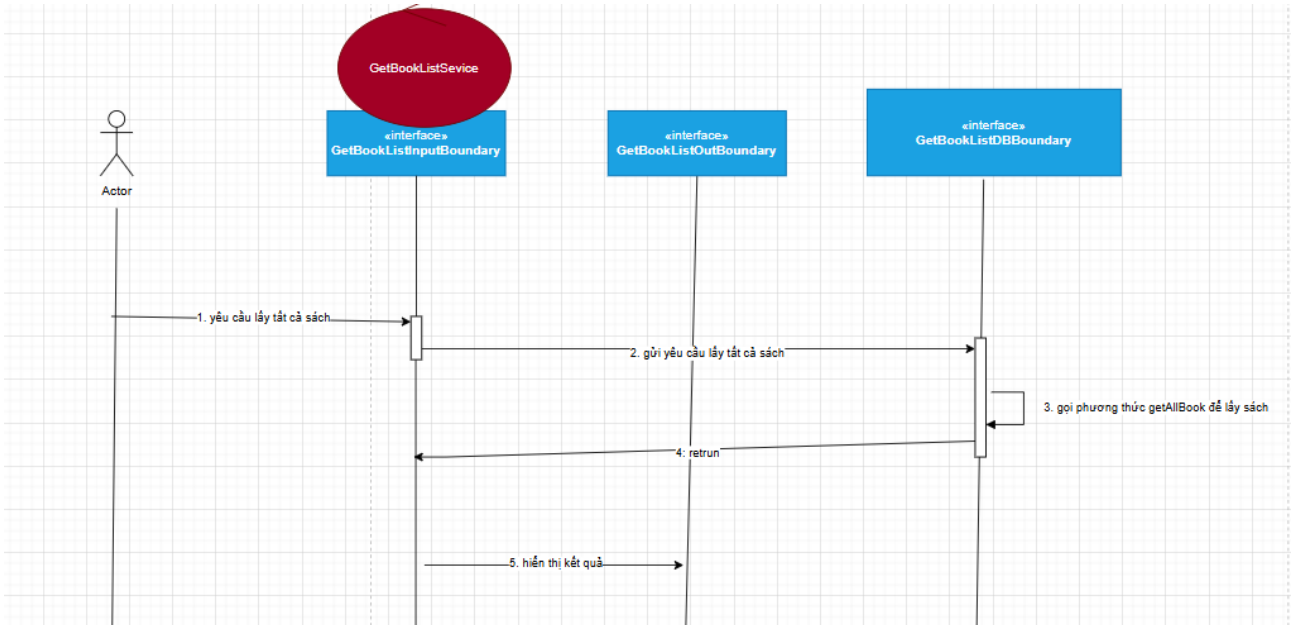
### Luồng hoạt động

- Actor gửi yêu cầu tính tổng giá trị sách thông qua giao diện TotalBookInputBoundary.

- Giao diện TotalBookInputBoundary chuyển tiếp yêu cầu đến lớp điều khiển TotalBookService.
- TotalBookService xử lý logic, kiểm tra thông tin liên quan đến đơn giá sách (ví dụ: đảm bảo dữ liệu đầu vào hợp lệ).
  - Nếu dữ liệu không hợp lệ, lớp điều khiển trả về thông báo lỗi qua giao diện TotalBookOutBoundary để hiển thị cho người dùng.
  - Nếu hợp lệ, lớp điều khiển tiếp tục gửi yêu cầu lấy dữ liệu đến cơ sở dữ liệu.
- Lớp điều khiển gửi yêu cầu qua giao diện TotalBookOutBoundary để truy xuất đơn giá sách từ cơ sở dữ liệu.
- Giao diện này gửi lệnh fetchPriceData() tới thành phần TotalBookDatabaseBoundary.
- TotalBookDatabaseBoundary truy xuất dữ liệu đơn giá từ cơ sở dữ liệu:
  - Nếu dữ liệu không tồn tại hoặc không thể truy xuất, cơ sở dữ liệu trả về thông báo lỗi.
  - Nếu truy xuất thành công, dữ liệu đơn giá được trả về.
- Lớp điều khiển nhận thông tin đơn giá, thực hiện tính toán tổng giá trị sách.
- Kết quả được chuyển qua giao diện TotalBookOutBoundary để gửi về cho người dùng.
- Giao diện hiển thị kết quả thông qua phương thức showResult(TotalPriceResult), thông báo tổng giá trị sách cho người dùng.



## 1.7 Sơ đồ tuần tự “GetBookList”



### Thành phần chính

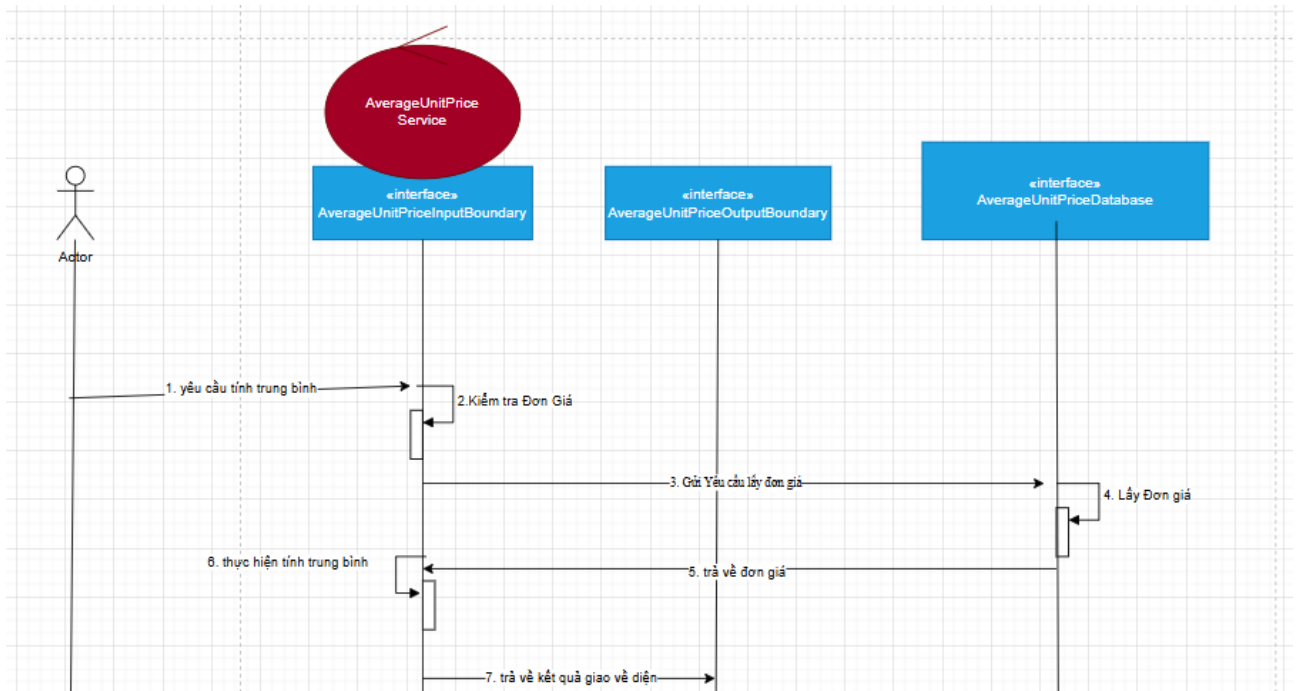
- **Actor:** Người yêu cầu lấy danh sách sách, có thể là người dùng, quản trị viên hoặc nhân viên thư viện.
- **GetBookListService:** Lớp điều khiển trung tâm, chịu trách nhiệm xử lý logic chính của quy trình lấy danh sách sách. Lớp này điều phối giữa các thành phần Input Boundary, Output Boundary và Database Boundary.
- **GetBookListInputBoundary:** Giao diện tiếp nhận yêu cầu lấy danh sách sách từ người dùng và chuyển thông tin đến lớp điều khiển chính.
- **GetBookListOutputBoundary:** Giao diện đảm nhận việc gửi yêu cầu lấy dữ liệu từ cơ sở dữ liệu và trả kết quả cho người dùng.
- **GetBookListDBBoundary:** Giao diện kết nối trực tiếp với cơ sở dữ liệu, thực hiện thao tác truy xuất thông tin danh sách sách.

### Lưuồng hoạt động

- Actor gửi yêu cầu lấy danh sách sách thông qua giao diện GetBookListInputBoundary.
- Giao diện GetBookListInputBoundary chuyển tiếp yêu cầu đến lớp điều khiển GetBookListService.

- GetBookListService xử lý logic và kiểm tra thông tin liên quan đến

### 1.8 Sơ đồ tuần tự “AverageUnitPrice”



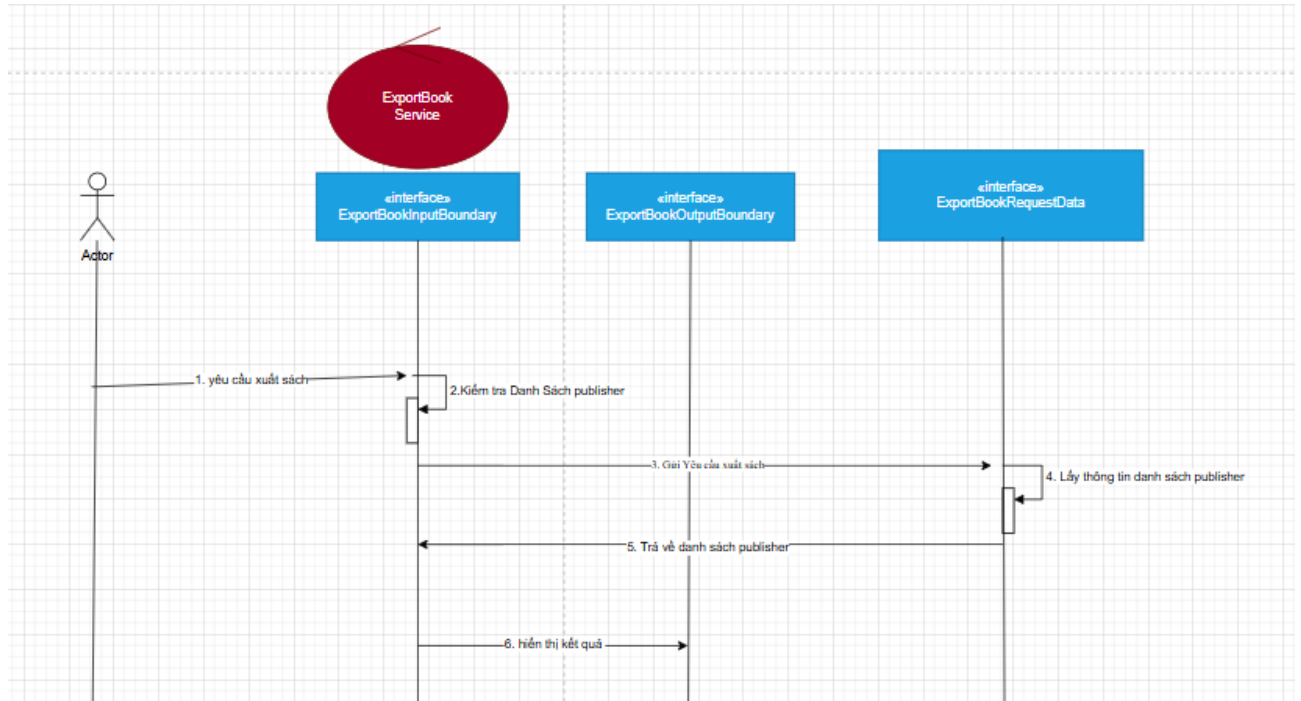
#### Thành phần chính

- **Actor:** Người yêu cầu tính đơn giá trung bình, có thể là quản trị viên hoặc nhân viên quản lý.
- **AverageUnitPriceService:** Lớp điều khiển trung tâm, chịu trách nhiệm xử lý logic chính của quy trình tính đơn giá trung bình. Lớp này điều phối giữa các thành phần Input Boundary, Output Boundary và Database Boundary.
- **AverageUnitPriceInputBoundary:** Giao diện tiếp nhận yêu cầu tính đơn giá trung bình từ người dùng và chuyển thông tin đến lớp điều khiển chính.
- **AverageUnitPriceOutputBoundary:** Giao diện đảm nhận việc gửi yêu cầu lấy dữ liệu từ cơ sở dữ liệu và trả kết quả tính toán cho người dùng.
- **AverageUnitPriceDatabase:** Giao diện kết nối trực tiếp với cơ sở dữ liệu, thực hiện thao tác truy xuất thông tin đơn giá.

## Luồng hoạt động

- Actor gửi yêu cầu tính đơn giá trung bình thông qua giao diện AverageUnitPriceInputBoundary.
- Giao diện AverageUnitPriceInputBoundary chuyển tiếp yêu cầu đến lớp điều khiển AverageUnitPriceService.
- AverageUnitPriceService xử lý logic, kiểm tra thông tin liên quan đến đơn giá (ví dụ: đảm bảo dữ liệu đầu vào hợp lệ).
  - Nếu dữ liệu không hợp lệ, lớp điều khiển trả về thông báo lỗi qua giao diện AverageUnitPriceOutputBoundary để hiển thị cho người dùng.
  - Nếu hợp lệ, lớp điều khiển tiếp tục gửi yêu cầu lấy dữ liệu đến cơ sở dữ liệu.
- Lớp điều khiển gửi yêu cầu qua giao diện AverageUnitPriceOutputBoundary để truy xuất đơn giá từ cơ sở dữ liệu.
- Giao diện này gửi lệnh lấy đơn giá tới thành phần AverageUnitPriceDatabase.
- AverageUnitPriceDatabase truy xuất dữ liệu đơn giá từ cơ sở dữ liệu:
  - Nếu dữ liệu không tồn tại hoặc không thể truy xuất, cơ sở dữ liệu trả về thông báo lỗi.
  - Nếu truy xuất thành công, dữ liệu đơn giá được trả về.
- Lớp điều khiển nhận thông tin đơn giá, thực hiện tính toán đơn giá trung bình.
- Kết quả được chuyển qua giao diện AverageUnitPriceOutputBoundary để gửi về cho người dùng.
- Giao diện hiển thị kết quả, thông báo đơn giá trung bình cho người dùng.

### 1.9 Sơ đồ tuần tự “ExportBook”



#### Thành phần chính

- **Actor:** Người yêu cầu xuất danh sách sách, có thể là quản trị viên hoặc nhân viên quản lý thư viện.
- **ExportBookService:** Lớp điều khiển trung tâm, chịu trách nhiệm xử lý logic chính của quy trình xuất danh sách sách. Lớp này điều phối giữa các thành phần Input Boundary, Output Boundary và Database Boundary.
- **ExportBookInputBoundary:** Giao diện tiếp nhận yêu cầu xuất danh sách từ người dùng và chuyển thông tin đến lớp điều khiển chính.
- **ExportBookOutputBoundary:** Giao diện đảm nhận việc gửi yêu cầu lấy dữ liệu từ cơ sở dữ liệu và trả kết quả cho người dùng.
- **ExportBookRequestData:** Giao diện kết nối trực tiếp với cơ sở dữ liệu, thực hiện thao tác truy xuất thông tin danh sách nhà xuất bản (publisher).

#### Luồng hoạt động

- Actor gửi yêu cầu xuất danh sách sách thông qua giao diện ExportBookInputBoundary.

- Giao diện ExportBookInputBoundary chuyển tiếp yêu cầu đến lớp điều khiển ExportBookService.
- ExportBookService xử lý logic, kiểm tra thông tin liên quan đến danh sách publisher (ví dụ: đảm bảo dữ liệu đầu vào hợp lệ).
  - Nếu dữ liệu không hợp lệ, lớp điều khiển trả về thông báo lỗi qua giao diện ExportBookOutputBoundary để hiển thị cho người dùng.
  - Nếu hợp lệ, lớp điều khiển tiếp tục gửi yêu cầu lấy dữ liệu đến cơ sở dữ liệu.
- Lớp điều khiển gửi yêu cầu qua giao diện ExportBookOutputBoundary để truy xuất danh sách publisher từ cơ sở dữ liệu.
- Giao diện này gửi lệnh lấy thông tin danh sách publisher tới thành phần ExportBookRequestData.
- ExportBookRequestData truy xuất dữ liệu danh sách publisher từ cơ sở dữ liệu:
  - Nếu dữ liệu không tồn tại hoặc không thể truy xuất, cơ sở dữ liệu trả về thông báo lỗi.
  - Nếu truy xuất thành công, dữ liệu danh sách publisher được trả về.
- Lớp điều khiển nhận thông tin danh sách publisher và trả về kết quả qua giao diện ExportBookOutputBoundary.
- Giao diện hiển thị kết quả danh sách publisher cho người dùng.

## 2. Sơ đồ class mức phân tích

## CHƯƠNG 3. KIỂM THỬ

### 1. Test case chức năng “addbook”

#### Trường hợp thêm sách TextBook với dữ liệu hợp lệ

- **Mô tả:** Kiểm tra xem dịch vụ có thể thêm một sách TextBook với dữ liệu hợp lệ vào repository và gửi kết quả qua output boundary đúng cách.
- **Mục tiêu:** Đảm bảo dịch vụ xử lý đúng logic thêm sách TextBook và không gặp lỗi.
- **Kỳ vọng:** Sách được lưu vào repository dưới dạng đối tượng TextBook, và kết quả được gửi đến AddBookOutputBoundary với thông điệp thành công.

#### Trường hợp thêm sách ReferenceBook với dữ liệu hợp lệ

- **Mô tả:** Kiểm tra xem dịch vụ có thể thêm một sách ReferenceBook với dữ liệu hợp lệ vào repository và gửi kết quả qua output boundary đúng cách.
- **Mục tiêu:** Đảm bảo dịch vụ xử lý đúng logic thêm sách ReferenceBook và không gặp lỗi.
- **Kỳ vọng:** Sách được lưu vào repository dưới dạng đối tượng ReferenceBook, và kết quả được gửi đến AddBookOutputBoundary với thông điệp thành công.

#### Trường hợp thêm sách với dữ liệu không hợp lệ (thiếu bookId)

- **Mô tả:** Kiểm tra khi dữ liệu thêm sách không hợp lệ (thiếu bookId), dịch vụ có thể phát hiện lỗi và gửi thông báo lỗi qua output boundary.
- **Mục tiêu:** Đảm bảo dịch vụ xử lý đúng lỗi dữ liệu đầu vào.
- **Kỳ vọng:** Kết quả được gửi đến AddBookOutputBoundary với thông báo lỗi.

### **Trường hợp thêm ReferenceBook với tax không hợp lệ**

- **Mô tả:** Kiểm tra khi thêm ReferenceBook với giá trị tax không hợp lệ (giá trị âm), dịch vụ có thể phát hiện lỗi và gửi thông báo lỗi qua output boundary.
- **Mục tiêu:** Đảm bảo dịch vụ phát hiện dữ liệu không hợp lệ và xử lý đúng cách.
- **Kỳ vọng:** Kết quả được gửi đến AddBookOutputBoundary với thông báo lỗi về giá trị tax.

### **Trường hợp thêm sách với ngày nhập không đúng định dạng**

- **Mô tả:** Kiểm tra khi thêm sách với giá trị ngày nhập không đúng định dạng, dịch vụ có thể phát hiện lỗi và gửi thông báo lỗi qua output boundary.
- **Mục tiêu:** Đảm bảo dịch vụ kiểm tra định dạng dữ liệu ngày nhập và xử lý đúng lỗi.
- **Kỳ vọng:** Kết quả được gửi đến AddBookOutputBoundary với thông báo lỗi về định dạng ngày.

### **Trường hợp thêm sách ReferenceBook nhưng thiếu giá trị tax**

- **Mô tả:** Kiểm tra khi thêm ReferenceBook mà không cung cấp giá trị tax, dịch vụ có thể phát hiện lỗi và gửi thông báo lỗi qua output boundary.
- **Mục tiêu:** Đảm bảo dịch vụ phát hiện dữ liệu thiếu sót và xử lý đúng cách.
- **Kỳ vọng:** Kết quả được gửi đến AddBookOutputBoundary với thông báo lỗi yêu cầu bổ sung giá trị tax.

### **Trường hợp thêm TextBook nhưng thiếu tình trạng (condition)**

- **Mô tả:** Kiểm tra khi thêm TextBook mà không cung cấp giá trị condition, dịch vụ có thể phát hiện lỗi và gửi thông báo lỗi qua output boundary.
- **Mục tiêu:** Đảm bảo dịch vụ phát hiện dữ liệu thiếu sót và xử lý đúng cách.
- **Kỳ vọng:** Kết quả được gửi đến AddBookOutputBoundary với thông báo lỗi yêu cầu bổ sung giá trị condition.

### **2. Test case chức năng “editbook”**

#### **Trường hợp chỉnh sửa sách thành công**

- **Mô tả:** Kiểm tra xem dịch vụ có thể chỉnh sửa sách với dữ liệu hợp lệ trong repository và gửi kết quả qua output boundary đúng cách.
- **Mục tiêu:** Đảm bảo dịch vụ thực hiện chỉnh sửa sách đúng logic và không xảy ra lỗi.
- **Kỳ vọng:** Sách được cập nhật thành công trong repository và kết quả được gửi đến EditBookOutputBoundary với thông điệp thành công.

#### **Trường hợp chỉnh sửa sách thất bại**

- **Mô tả:** Kiểm tra khi quá trình chỉnh sửa sách thất bại (ví dụ: sách không tồn tại hoặc xảy ra lỗi trong repository), dịch vụ có xử lý đúng cách và gửi thông báo lỗi qua output boundary.
- **Mục tiêu:** Đảm bảo dịch vụ phát hiện lỗi trong quá trình chỉnh sửa và xử lý đúng cách.
- **Kỳ vọng:** Dịch vụ phát hiện lỗi, không gây ra ngoại lệ không mong muốn, và kết quả được gửi đến EditBookOutputBoundary với thông báo lỗi.



### **Trường hợp chỉnh sửa sách với dữ liệu không hợp lệ (thiếu bookId)**

- Mô tả: Kiểm tra khi dữ liệu đầu vào không hợp lệ (thiếu bookId), dịch vụ có thể phát hiện lỗi và không thực hiện chỉnh sửa trong repository.
- Mục tiêu: Đảm bảo dịch vụ kiểm tra dữ liệu đầu vào và xử lý đúng các trường hợp dữ liệu không hợp lệ.
- Kỳ vọng: Phương thức editBook trong repository không được gọi, và kết quả được gửi đến EditBookOutputBoundary với thông báo lỗi về dữ liệu không hợp lệ.

### **Trường hợp chỉnh sửa sách nhưng ngày không đúng định dạng**

- Mô tả: Kiểm tra khi ngày truyền vào không đúng định dạng, dịch vụ có thể phát hiện lỗi và gửi thông báo lỗi qua output boundary.
- Mục tiêu: Đảm bảo dịch vụ kiểm tra định dạng dữ liệu ngày và xử lý đúng lỗi.
- Kỳ vọng: Phương thức editBook trong repository không được gọi, và kết quả được gửi đến EditBookOutputBoundary với thông báo lỗi về định dạng ngày.

### **Trường hợp chỉnh sửa sách nhưng giá trị price âm**

- Mô tả: Kiểm tra khi dữ liệu chỉnh sửa có giá trị price âm, dịch vụ có thể phát hiện lỗi và gửi thông báo lỗi qua output boundary.
- Mục tiêu: Đảm bảo dịch vụ phát hiện dữ liệu không hợp lệ và xử lý đúng cách.
- Kỳ vọng: Phương thức editBook trong repository không được gọi, và kết quả được gửi đến EditBookOutputBoundary với thông báo lỗi về giá trị price.

### **Trường hợp chỉnh sửa sách nhưng bookId không tồn tại trong repository**

- **Mô tả:** Kiểm tra khi bookId không tồn tại trong repository, dịch vụ có thể phát hiện lỗi và gửi thông báo lỗi qua output boundary.
- **Mục tiêu:** Đảm bảo dịch vụ xử lý đúng lỗi khi sách cần chỉnh sửa không tồn tại.
- **Kỳ vọng:** Không có thay đổi nào được thực hiện trong repository, và kết quả được gửi đến EditBookOutputBoundary với thông báo lỗi rằng sách không tồn tại.

### **Trường hợp chỉnh sửa sách nhưng repository trả về null**

- **Mô tả:** Kiểm tra khi repository trả về giá trị null trong quá trình chỉnh sửa, dịch vụ có xử lý đúng cách và tránh lỗi ngoại lệ.
- **Mục tiêu:** Đảm bảo dịch vụ xử lý đúng các lỗi từ repository.
- **Kỳ vọng:** Không có ngoại lệ không mong muốn xảy ra, và kết quả được gửi đến EditBookOutputBoundary với thông báo lỗi thích hợp.

## **3. Test case chức năng “deletebook”**

### **Trường hợp xóa sách thành công**

- **Mô tả:** Kiểm tra khi yêu cầu xóa sách có dữ liệu hợp lệ và sách tồn tại trong repository, dịch vụ có thể xóa sách đúng cách và gửi kết quả qua output boundary.
- **Mục tiêu:** Đảm bảo sách được xóa thành công khỏi repository và không xảy ra lỗi.
- **Kỳ vọng:** Phương thức deleteBook của repository được gọi và trả về true. Kết quả được gửi đến DeleteBookOutputBoundary với thông báo thành công.

### **Trường hợp xóa sách thất bại**

- **Mô tả:** Kiểm tra khi yêu cầu xóa sách có dữ liệu hợp lệ nhưng xảy ra lỗi trong repository, dịch vụ có xử lý đúng cách và gửi thông báo lỗi qua output boundary.
- **Mục tiêu:** Đảm bảo dịch vụ phát hiện lỗi và xử lý đúng khi không thể xóa sách.
- **Kỳ vọng:** Phương thức deleteBook của repository được gọi và trả về false. Kết quả được gửi đến DeleteBookOutputBoundary với thông báo thất bại.

### **Trường hợp xóa sách không tồn tại trong repository**

- **Mô tả:** Kiểm tra khi sách cần xóa không tồn tại trong repository, dịch vụ có thể phát hiện lỗi và gửi thông báo lỗi qua output boundary.
- **Mục tiêu:** Đảm bảo dịch vụ phát hiện sách không tồn tại và xử lý đúng cách.
- **Kỳ vọng:** Phương thức deleteBook của repository được gọi và trả về false. Kết quả được gửi đến DeleteBookOutputBoundary với thông báo sách không tồn tại.

### **Trường hợp xóa sách với bookId rỗng**

- **Mô tả:** Kiểm tra khi dữ liệu đầu vào không hợp lệ (không có bookId), dịch vụ có thể phát hiện lỗi và không thực hiện bất kỳ hành động nào trong repository.
- **Mục tiêu:** Đảm bảo dịch vụ kiểm tra dữ liệu đầu vào và xử lý đúng khi bookId bị thiếu hoặc rỗng.

- **Kỳ vọng:** Phương thức deleteBook trong repository không được gọi, và kết quả được gửi đến DeleteBookOutputBoundary với thông báo lỗi về dữ liệu không hợp lệ.

### **Trường hợp repository trả về null**

- **Mô tả:** Kiểm tra khi repository trả về giá trị null trong quá trình xóa sách, dịch vụ có xử lý đúng cách và tránh lỗi ngoại lệ.
- **Mục tiêu:** Đảm bảo dịch vụ xử lý đúng các lỗi từ repository.
- **Kỳ vọng:** Không có ngoại lệ không mong muốn xảy ra. Kết quả được gửi đến DeleteBookOutputBoundary với thông báo lỗi thích hợp.

### **Trường hợp xóa sách nhưng repository gặp lỗi (RuntimeException)**

- **Mô tả:** Kiểm tra khi repository ném ra ngoại lệ trong quá trình xóa sách, dịch vụ có thể xử lý đúng cách và gửi thông báo lỗi qua output boundary.
- **Mục tiêu:** Đảm bảo dịch vụ không bị crash và gửi thông báo lỗi thích hợp khi xảy ra ngoại lệ.
- **Kỳ vọng:** Dịch vụ phát hiện ngoại lệ, không gây ra crash, và kết quả được gửi đến DeleteBookOutputBoundary với thông báo lỗi.

### **Trường hợp xóa sách nhưng repository bị mất kết nối**

- **Mô tả:** Kiểm tra khi repository không thể kết nối (giả lập trường hợp mất kết nối cơ sở dữ liệu), dịch vụ có thể phát hiện và xử lý đúng cách.
- **Mục tiêu:** Đảm bảo dịch vụ xử lý lỗi mất kết nối cơ sở dữ liệu một cách an toàn.

- **Kỳ vọng:** Phương thức deleteBook trong repository không thành công, và kết quả được gửi đến DeleteBookOutputBoundary với thông báo lỗi kết nối.

#### 4. Test case chức năng “printbook”

##### Trường hợp danh sách sách có dữ liệu hợp lệ

- **Mô tả:** Kiểm tra khi repository trả về danh sách sách hợp lệ, dịch vụ có thể xử lý và gửi dữ liệu đúng qua output boundary.
- **Mục tiêu:** Đảm bảo tất cả sách hợp lệ trong danh sách được lấy ra và truyền đúng cách đến PrintBookOutputBoundary.
- **Kỳ vọng:** Phương thức getAllBooks trong repository được gọi, và presentPrintBookResult được gọi với danh sách sách hợp lệ.

##### Trường hợp danh sách sách trống

- **Mô tả:** Kiểm tra khi repository không chứa sách nào, dịch vụ có xử lý đúng và không xảy ra lỗi.
- **Mục tiêu:** Đảm bảo dịch vụ có thể xử lý danh sách trống và không phát sinh lỗi khi không có sách nào.
- **Kỳ vọng:** Phương thức getAllBooks được gọi, và presentPrintBookResult được gọi với danh sách trống.

##### Trường hợp repository trả về null

- **Mô tả:** Kiểm tra khi repository trả về giá trị null, dịch vụ có thể xử lý đúng và tránh lỗi ngoại lệ.
- **Mục tiêu:** Đảm bảo dịch vụ phát hiện null từ repository và xử lý bằng cách gửi danh sách trống đến output boundary.

- **Kỳ vọng:** Phương thức `getAllBooks` được gọi, và `presentPrintBookResult` được gọi với danh sách trống.
- 

### **Trường hợp danh sách sách chứa đối tượng null**

- **Mô tả:** Kiểm tra khi danh sách sách từ repository chứa một hoặc nhiều đối tượng null, dịch vụ có thể bỏ qua các đối tượng này và xử lý phần còn lại đúng cách.
  - **Mục tiêu:** Đảm bảo dịch vụ không gặp lỗi khi xử lý danh sách sách chứa null và chỉ gửi các đối tượng hợp lệ qua output boundary.
  - **Kỳ vọng:** Phương thức `getAllBooks` được gọi, và `presentPrintBookResult` được gọi với danh sách đã lọc bỏ các đối tượng null.
- 

### **Trường hợp repository gặp lỗi (RuntimeException)**

- **Mô tả:** Kiểm tra khi repository gặp lỗi trong quá trình lấy danh sách sách, dịch vụ có thể xử lý đúng cách và tránh bị crash.
  - **Mục tiêu:** Đảm bảo dịch vụ phát hiện lỗi và xử lý một cách an toàn, gửi thông báo lỗi thích hợp qua output boundary.
  - **Kỳ vọng:** Dịch vụ không bị crash và phương thức `presentPrintBookResult` được gọi với danh sách trống hoặc thông báo lỗi.
- 

### **Trường hợp danh sách sách có nhiều loại sách khác nhau**

- **Mô tả:** Kiểm tra khi repository trả về danh sách sách chứa nhiều loại sách khác nhau (sách tham khảo và sách giáo khoa), dịch vụ có thể xử lý tất cả các loại sách đúng cách.
- **Mục tiêu:** Đảm bảo dịch vụ có thể lấy thông tin từ nhiều loại sách khác nhau trong repository và gửi qua output boundary mà không xảy ra lỗi.

- **Kỳ vọng:** Phương thức `getAllBooks` được gọi, và `presentPrintBookResult` được gọi với danh sách sách có đầy đủ các loại sách.
- 

#### **Trường hợp danh sách sách chứa sách trùng lặp**

- **Mô tả:** Kiểm tra khi danh sách sách từ repository chứa các đối tượng sách trùng lặp, dịch vụ có thể xử lý đúng cách mà không gây lỗi.
  - **Mục tiêu:** Đảm bảo dịch vụ có thể xử lý các sách trùng lặp mà không gây ra lỗi hoặc mất thông tin.
  - **Kỳ vọng:** Phương thức `getAllBooks` được gọi, và `presentPrintBookResult` được gọi với danh sách chứa sách trùng lặp.
- 

#### **Trường hợp repository mất kết nối**

- **Mô tả:** Kiểm tra khi repository không thể kết nối (mất kết nối cơ sở dữ liệu), dịch vụ có thể phát hiện lỗi và xử lý đúng cách.
- **Mục tiêu:** Đảm bảo dịch vụ không gặp lỗi nghiêm trọng khi repository mất kết nối.
- **Kỳ vọng:** Phương thức `presentPrintBookResult` được gọi với danh sách trống hoặc thông báo lỗi.

### **5. Test case chức năng “searchbook”**

#### **Tìm kiếm sách TextBook với bookId hợp lệ**

- **Mô tả:** Kiểm tra khi bookId hợp lệ, repository trả về một danh sách chứa TextBook, và dịch vụ có thể xử lý đúng cách.
- **Mục tiêu:** Đảm bảo sách TextBook hợp lệ được tìm kiếm và kết quả được truyền đúng đến `SearchBookOutputBoundary`.
- **Kỳ vọng:**

- Phương thức `searchBooksById` của repository được gọi với đúng `bookId`.
  - Phương thức `presentSearchBookResults` của output boundary được gọi với danh sách `TextBook` hợp lệ.
- 

### **Tìm kiếm sách `ReferenceBook` với `bookId` hợp lệ**

- **Mô tả:** Kiểm tra khi `bookId` hợp lệ, repository trả về một danh sách chứa `ReferenceBook`, và dịch vụ có thể xử lý đúng cách.
  - **Mục tiêu:** Đảm bảo sách `ReferenceBook` hợp lệ được tìm kiếm và kết quả được truyền đúng đến `SearchBookOutputBoundary`.
  - **Kỳ vọng:**
    - Phương thức `searchBooksById` của repository được gọi với đúng `bookId`.
    - Phương thức `presentSearchBookResults` của output boundary được gọi với danh sách `ReferenceBook` hợp lệ.
- 

### **Tìm kiếm với `bookId` không hợp lệ**

- **Mô tả:** Kiểm tra khi `bookId` không hợp lệ, repository trả về danh sách rỗng, và dịch vụ xử lý đúng.
- **Mục tiêu:** Đảm bảo không có lỗi xảy ra khi `bookId` không hợp lệ và danh sách kết quả rỗng được truyền đến `SearchBookOutputBoundary`.
- **Kỳ vọng:**
  - Phương thức `searchBooksById` của repository được gọi với `bookId` không hợp lệ.
  - Phương thức `presentSearchBookResults` của output boundary được gọi với danh sách rỗng.



---

### **Tìm kiếm khi repository trả về null**

- **Mô tả:** Kiểm tra trường hợp searchBooksById của repository trả về giá trị null, dịch vụ xử lý đúng cách mà không gặp lỗi.
- **Mục tiêu:** Đảm bảo dịch vụ phát hiện null và xử lý bằng cách truyền một danh sách rỗng đến SearchBookOutputBoundary.
- **Kỳ vọng:**
  - Phương thức searchBooksById của repository được gọi.
  - Phương thức presentSearchBookResults của output boundary được gọi với danh sách rỗng.

---

### **Tìm kiếm với danh sách sách chứa đối tượng null**

- **Mô tả:** Kiểm tra khi danh sách trả về từ repository chứa một hoặc nhiều đối tượng null, dịch vụ bỏ qua các đối tượng này và xử lý phần còn lại đúng cách.
- **Mục tiêu:** Đảm bảo danh sách sách hợp lệ được xử lý và các đối tượng null bị loại bỏ.
- **Kỳ vọng:**
  - Phương thức searchBooksById của repository được gọi.
  - Phương thức presentSearchBookResults của output boundary được gọi với danh sách đã loại bỏ các đối tượng null.

---

### **Tìm kiếm khi repository gặp lỗi (RuntimeException)**

- **Mô tả:** Kiểm tra khi repository gặp lỗi (ví dụ: không thể kết nối cơ sở dữ liệu), dịch vụ phát hiện lỗi và xử lý an toàn.

- **Mục tiêu:** Đảm bảo dịch vụ không bị crash và gửi thông báo lỗi hoặc danh sách rỗng đến SearchBookOutputBoundary.
  - **Kỳ vọng:**
    - Phương thức presentSearchBookResults của output boundary được gọi với danh sách rỗng hoặc thông báo lỗi.
- 

### **Đảm bảo phương thức searchBooksById của repository được gọi đúng cách**

- **Mô tả:** Kiểm tra dịch vụ có gọi đúng phương thức searchBooksById của repository với tham số chính xác.
  - **Mục tiêu:** Đảm bảo bookId được truyền chính xác vào repository.
  - **Kỳ vọng:**
    - Phương thức searchBooksById của repository được gọi với đúng bookId.
- 

### **Tìm kiếm khi danh sách sách có dữ liệu trùng lặp**

- **Mô tả:** Kiểm tra khi danh sách trả về từ repository chứa các sách trùng lặp, dịch vụ có thể xử lý mà không bị lỗi.
  - **Mục tiêu:** Đảm bảo các sách trùng lặp không gây lỗi hoặc ảnh hưởng đến kết quả tìm kiếm.
  - **Kỳ vọng:**
    - Phương thức searchBooksById được gọi.
    - Phương thức presentSearchBookResults được gọi với danh sách chứa sách trùng lặp.
- 

### **Tìm kiếm khi bookId là chuỗi rỗng**

- **Mô tả:** Kiểm tra khi bookId được truyền vào là chuỗi rỗng, dịch vụ xử lý và trả về danh sách rỗng.
- **Mục tiêu:** Đảm bảo dịch vụ phát hiện bookId không hợp lệ và xử lý an toàn.
- **Kỳ vọng:**
  - Phương thức searchBooksById không trả về lỗi.
  - Phương thức presentSearchBookResults được gọi với danh sách rỗng.

## 6. Test case chức năng “totalbookprice”

### Tính tổng giá sách với danh sách sách hợp lệ

- **Mô tả:** Kiểm tra trường hợp danh sách sách hợp lệ từ repository (bao gồm cả TextBook và ReferenceBook) được tính toán chính xác.
- **Mục tiêu:**
  - Đảm bảo phương thức calculateTotalPrice() được gọi trên từng đối tượng sách.
  - Đảm bảo tổng giá sách được tính đúng và truyền tới TotalBookPriceOutputBoundary.
- **Kỳ vọng:**
  - Phương thức getAllBooks() của repository được gọi.
  - Phương thức calculateTotalPrice() được gọi cho từng sách.
  - Phương thức presentTotalBookPrice() của output boundary được gọi với tổng giá trị chính xác.

---

### Tính tổng giá sách khi danh sách sách rỗng

- **Mô tả:** Kiểm tra trường hợp repository trả về danh sách sách rỗng.

- **Mục tiêu:**
    - Đảm bảo tổng giá trị trả về là 0.0 và không có lỗi.
    - Đảm bảo output boundary nhận được tổng giá trị là 0.0.
  - **Kỳ vọng:**
    - Phương thức getAllBooks() của repository được gọi.
    - Phương thức presentTotalBookPrice() của output boundary được gọi với tổng giá trị 0.0.
- 

### **Tính tổng giá sách khi repository trả về null**

- **Mô tả:** Kiểm tra trường hợp repository trả về null.
  - **Mục tiêu:**
    - Đảm bảo dịch vụ phát hiện null và xử lý bằng cách coi như danh sách rỗng.
    - Đảm bảo output boundary nhận được tổng giá trị là 0.0.
  - **Kỳ vọng:**
    - Phương thức getAllBooks() của repository được gọi.
    - Phương thức presentTotalBookPrice() của output boundary được gọi với tổng giá trị 0.0.
- 

### **Tính tổng giá sách khi có sách với giá trị 0.0**

- **Mô tả:** Kiểm tra khi danh sách sách chứa các sách có tổng giá trị là 0.0 (do tính toán trả về 0.0).
- **Mục tiêu:**
  - Đảm bảo tổng giá trị được tính chính xác.
  - Đảm bảo output boundary nhận được tổng giá trị đúng (là 0.0).

- **Kỳ vọng:**
    - Phương thức `getAllBooks()` được gọi.
    - Phương thức `presentTotalBookPrice()` của `output boundary` được gọi với tổng giá trị 0.0.
- 

### **Tính tổng giá sách khi có sách với giá trị âm**

- **Mô tả:** Kiểm tra khi danh sách sách chứa các sách có tổng giá trị âm (do tính toán không chính xác hoặc dữ liệu lỗi).
  - **Mục tiêu:**
    - Đảm bảo tổng giá trị được tính toán đúng hoặc được kiểm tra hợp lệ trước khi trả về.
    - Đảm bảo `output boundary` nhận được tổng giá trị hợp lệ (không phải giá trị âm).
  - **Kỳ vọng:**
    - Phương thức `calculateTotalPrice()` trả về giá trị hợp lệ.
    - Phương thức `presentTotalBookPrice()` được gọi với tổng giá trị hợp lệ.
- 

### **Tính tổng giá sách khi danh sách sách chứa đối tượng null**

- **Mô tả:** Kiểm tra trường hợp danh sách sách trả về từ repository chứa một hoặc nhiều đối tượng null.
- **Mục tiêu:**
  - Đảm bảo dịch vụ phát hiện và bỏ qua các đối tượng null.
  - Đảm bảo tổng giá trị được tính toán đúng dựa trên các sách hợp lệ.
- **Kỳ vọng:**

- Phương thức `calculateTotalPrice()` không được gọi trên các đối tượng null.
- Phương thức `presentTotalBookPrice()` được gọi với tổng giá trị đúng.

---

### **Tính tổng giá sách khi repository gặp lỗi (RuntimeException)**

- **Mô tả:** Kiểm tra khi repository gặp lỗi (ví dụ: không thể kết nối cơ sở dữ liệu).
- **Mục tiêu:**
  - Đảm bảo dịch vụ phát hiện lỗi và xử lý an toàn (trả về tổng giá trị là 0.0).
  - Đảm bảo output boundary nhận được kết quả hợp lệ.
- **Kỳ vọng:**
  - Phương thức `getAllBooks()` được gọi và xử lý lỗi.
  - Phương thức `presentTotalBookPrice()` của output boundary được gọi với tổng giá trị 0.0.

---

### **Tính tổng giá sách khi danh sách sách chứa sách trùng lặp**

- **Mô tả:** Kiểm tra khi danh sách sách trả về từ repository chứa các sách trùng lặp.
- **Mục tiêu:**
  - Đảm bảo tổng giá trị được tính đúng (bao gồm tất cả các sách trùng lặp).
  - Đảm bảo không có lỗi khi xử lý danh sách trùng lặp.
- **Kỳ vọng:**
  - Phương thức `getAllBooks()` được gọi.

- Phương thức `presentTotalBookPrice()` của `output boundary` được gọi với tổng giá trị đúng.

---

### **Tính tổng giá sách khi danh sách sách có nhiều loại khác nhau**

- **Mô tả:** Kiểm tra trường hợp danh sách sách trả về từ repository chứa cả `TextBook` và `ReferenceBook`.
- **Mục tiêu:**
  - Đảm bảo tổng giá trị được tính đúng cho cả hai loại sách.
  - Đảm bảo `output boundary` nhận được tổng giá trị đúng.
- **Kỳ vọng:**
  - Phương thức `calculateTotalPrice()` được gọi trên tất cả các sách.
  - Phương thức `presentTotalBookPrice()` được gọi với tổng giá trị đúng.

### **7. Test case chức năng “averageunitprice”**

#### **Tính giá trung bình với danh sách sách tham chiếu hợp lệ**

- **Mô tả:** Kiểm tra trường hợp có ít nhất một sách tham chiếu (`ReferenceBook`) trong danh sách trả về từ repository.
- **Mục tiêu:**
  - Đảm bảo phương thức `getUnitPrice()` được gọi trên từng đối tượng `ReferenceBook`.
  - Đảm bảo giá trung bình được tính toán đúng và truyền tới `AverageUnitPriceOutputBoundary`.
- **Kỳ vọng:**
  - Phương thức `getAllBooks()` của repository được gọi.

- Phương thức `getUnitPrice()` được gọi trên từng đối tượng `ReferenceBook`.
  - Phương thức `presentAverageUnitPriceResult()` của output boundary được gọi với giá trung bình chính xác.
- 

### **Tính giá trung bình khi không có sách tham chiếu**

- **Mô tả:** Kiểm tra trường hợp danh sách sách trả về từ repository không chứa bất kỳ sách tham chiếu nào.
  - **Mục tiêu:**
    - Đảm bảo giá trung bình trả về là 0.0 và không có lỗi.
    - Đảm bảo output boundary nhận được giá trị 0.0.
  - **Kỳ vọng:**
    - Phương thức `getAllBooks()` của repository được gọi.
    - Phương thức `presentAverageUnitPriceResult()` được gọi với giá trị trung bình 0.0.
- 

### **Tính giá trung bình khi không có sách trong repository**

- **Mô tả:** Kiểm tra trường hợp repository trả về danh sách sách rỗng.
- **Mục tiêu:**
  - Đảm bảo giá trung bình trả về là 0.0 khi danh sách sách rỗng.
  - Đảm bảo output boundary nhận được giá trị 0.0.
- **Kỳ vọng:**
  - Phương thức `getAllBooks()` của repository được gọi.
  - Phương thức `presentAverageUnitPriceResult()` được gọi với giá trị trung bình 0.0.



---

### **Tính giá trung bình với một sách tham chiếu duy nhất**

- **Mô tả:** Kiểm tra trường hợp danh sách sách chứa duy nhất một đối tượng ReferenceBook.
- **Mục tiêu:**
  - Đảm bảo giá trung bình chính bằng giá trị `getUnitPrice()` của sách duy nhất.
  - Đảm bảo output boundary nhận được giá trị trung bình chính xác.
- **Kỳ vọng:**
  - Phương thức `getUnitPrice()` của ReferenceBook được gọi.
  - Phương thức `presentAverageUnitPriceResult()` được gọi với giá trị trung bình đúng.

---

### **Tính giá trung bình khi repository trả về null**

- **Mô tả:** Kiểm tra trường hợp repository trả về null.
- **Mục tiêu:**
  - Đảm bảo dịch vụ xử lý tình huống null và coi như danh sách rỗng.
  - Đảm bảo giá trị trung bình trả về là 0.0.
- **Kỳ vọng:**
  - Phương thức `getAllBooks()` của repository được gọi.
  - Phương thức `presentAverageUnitPriceResult()` được gọi với giá trị trung bình 0.0.

---

### **Tính giá trung bình với sách trùng lặp**

- **Mô tả:** Kiểm tra trường hợp danh sách sách trả về từ repository chứa các sách tham chiếu bị trùng lặp.
  - **Mục tiêu:**
    - Đảm bảo giá trung bình được tính đúng dựa trên danh sách chứa các sách trùng lặp.
    - Đảm bảo không có lỗi khi xử lý sách trùng lặp.
  - **Kỳ vọng:**
    - Phương thức `getAllBooks()` được gọi.
    - Phương thức `presentAverageUnitPriceResult()` được gọi với giá trị trung bình đúng.
- 

### **Tính giá trung bình khi tất cả giá sách là 0.0**

- **Mô tả:** Kiểm tra trường hợp tất cả sách tham chiếu trong danh sách có giá trị `getUnitPrice()` là 0.0.
  - **Mục tiêu:**
    - Đảm bảo giá trị trung bình trả về là 0.0.
    - Đảm bảo output boundary nhận được giá trị 0.0.
  - **Kỳ vọng:**
    - Phương thức `getUnitPrice()` được gọi trên từng `ReferenceBook`.
    - Phương thức `presentAverageUnitPriceResult()` được gọi với giá trị trung bình 0.0.
- 

### **Tính giá trung bình khi có sách không phải `ReferenceBook`**

- **Mô tả:** Kiểm tra khi danh sách trả về từ repository chứa các sách không phải loại `ReferenceBook`.
- **Mục tiêu:**

- Đảm bảo chỉ tính giá trung bình trên các sách tham chiếu (ReferenceBook).
  - Bỏ qua các sách không hợp lệ.
  - **Kỳ vọng:**
    - Phương thức `getUnitPrice()` không được gọi trên sách không phải ReferenceBook.
    - Phương thức `presentAverageUnitPriceResult()` được gọi với giá trị trung bình đúng.
- 

### **Tính giá trung bình khi repository gặp lỗi**

- **Mô tả:** Kiểm tra trường hợp repository gặp lỗi (ví dụ: không thể kết nối cơ sở dữ liệu).
  - **Mục tiêu:**
    - Đảm bảo dịch vụ phát hiện lỗi và xử lý an toàn (trả về giá trị trung bình là 0.0).
    - Đảm bảo output boundary nhận được kết quả hợp lệ.
  - **Kỳ vọng:**
    - Phương thức `getAllBooks()` được gọi và xử lý lỗi.
    - Phương thức `presentAverageUnitPriceResult()` được gọi với giá trị trung bình 0.0.
- 

### **Tính giá trung bình khi có nhiều sách tham chiếu với giá trị khác nhau**

- **Mô tả:** Kiểm tra khi danh sách sách chứa nhiều ReferenceBook với giá trị `getUnitPrice()` khác nhau.
- **Mục tiêu:**

- Đảm bảo giá trung bình được tính chính xác dựa trên tất cả sách tham chiếu.
- Đảm bảo output boundary nhận được giá trị đúng.
- **Kỳ vọng:**
  - Phương thức `getUnitPrice()` được gọi trên tất cả các `ReferenceBook`.
  - Phương thức `presentAverageUnitPriceResult()` được gọi với giá trị trung bình chính xác.

## CHƯƠNG 4: KẾT QUẢ

### 1. Đánh giá phần mềm

Phần mềm được đánh giá dựa trên các tiêu chí về mục đích, phạm vi, tính năng, hiệu suất, và tính khả dụng. Trước hết, phần mềm đã đáp ứng tốt các yêu cầu ban đầu, ví dụ như quản lý thông tin sách, tìm kiếm và tính toán giá trị liên quan, tuy nhiên vẫn còn thiếu một số tính năng nâng cao như phân tích xu hướng mượn sách để hỗ trợ các quyết định chiến lược. Về mặt tính năng, các chức năng chính được triển khai hiệu quả và đáp ứng nhu cầu cơ bản của người dùng, nhưng một số phần giao diện và trải nghiệm người dùng có thể được cải thiện để dễ sử dụng hơn, đặc biệt với người dùng mới.

Hiệu suất của phần mềm ổn định với các tác vụ cơ bản, tuy nhiên khi xử lý dữ liệu lớn, hệ thống có dấu hiệu chậm lại, điều này cần được tối ưu. Phần mềm cũng được thiết kế linh hoạt để hỗ trợ mở rộng trong tương lai, một điểm cộng quan trọng cho các yêu cầu phát triển lâu dài.

Tóm lại, phần mềm đã hoàn thành tốt nhiệm vụ chính, nhưng để trở thành một sản phẩm tối ưu, cần tập trung cải tiến hiệu suất, giao diện và bổ sung thêm các tính năng nâng cao.

## 2. Hình ảnh tất cả các chức năng của chương trình

### 2.1. Chức năng lấy thông tin tất cả sách

Mã Sách	Ngày Nhập	Đơn Giá	Số Lượng	Nhà Xuất Bản	Thuế	Tình Trạng
GT001	2024-10-01	150000.0	50	NXB Giáo Dục		Mới
GT002	2024-10-02	180000.0	60	NXB Giáo Dục		Mới
GT003	2024-10-03	120000.0	40	NXB Đại Học		Cũ
GT004	2024-10-04	160000.0	45	NXB Đại Học		Mới
GT005	2024-10-05	140000.0	55	NXB Giáo Dục		Cũ
GT006	2024-10-06	175000.0	50	NXB Đại Học		Mới
GT007	2024-10-07	165000.0	60	NXB Giáo Dục		Cũ
GT008	2024-10-08	150000.0	50	NXB Đại Học		Mới
GT009	2024-10-09	120000.0	55	NXB Giáo Dục		Cũ
GT010	2024-10-10	130000.0	60	NXB Đại Học		Mới
GT011	2024-10-11	155000.0	50	NXB Giáo Dục		Mới
GT012	2024-10-12	145000.0	55	NXB Đại Học		Cũ
GT013	2024-10-13	125000.0	50	NXB Giáo Dục		Mới
GT014	2024-10-14	160000.0	40	NXB Đại Học		Cũ
GT015	2024-10-15	170000.0	60	NXB Giáo Dục		Mới
GT016	2024-10-16	145000.0	50	NXB Đại Học		Mới
GT017	2024-10-17	155000.0	55	NXB Giáo Dục		Cũ
GT018	2024-10-18	130000.0	60	NXB Đại Học		Mới
GT019	2024-10-19	140000.0	50	NXB Giáo Dục		Mới
GT020	2024-10-20	160000.0	55	NXB Đại Học		Cũ
RF001	2024-10-01	250000.0	30	NXB Khoa Học	10.0	
RF002	2024-10-02	300000.0	25	NXB Khoa Học	12.0	
RF003	2024-10-03	220000.0	40	NXB Kinh Tế	8.0	
RF004	2024-10-04	280000.0	35	NXB Kinh Tế	15.0	
RF005	2024-10-05	240000.0	50	NXB Khoa Học	9.0	
RF006	2024-10-06	310000.0	30	NXB Kinh Tế	13.0	
RF007	2024-10-07	330000.0	45	NXB Khoa Học	14.0	
RF008	2024-10-08	250000.0	40	NXB Kinh Tế	10.0	
RF009	2024-10-09	270000.0	60	NXB Khoa Học	11.0	
RF010	2024-10-10	290000.0	30	NXB Kinh Tế	10.0	
RF011	2024-10-11	310000.0	50	NXB Khoa Học	13.0	
RF012	2024-10-12	260000.0	40	NXB Kinh Tế	9.0	
RF013	2024-10-13	240000.0	45	NXB Khoa Học	8.0	
RF014	2024-10-14	300000.0	60	NXB Kinh Tế	12.0	
RF015	2024-10-15	320000.0	25	NXB Khoa Học	14.0	
RF016	2024-10-16	250000.0	35	NXB Kinh Tế	11.0	
RF017	2024-10-17	230000.0	50	NXB Khoa Học	10.0	
RF018	2024-10-18	290000.0	30	NXB Kinh Tế	12.0	
RF019	2024-10-19	260000.0	40	NXB Khoa Học	9.0	
RF020	2024-10-20	280000.0	45	NXB Kinh Tế	13.0	

Hình 1.1 Chức năng lấy thông tin tất cả sách

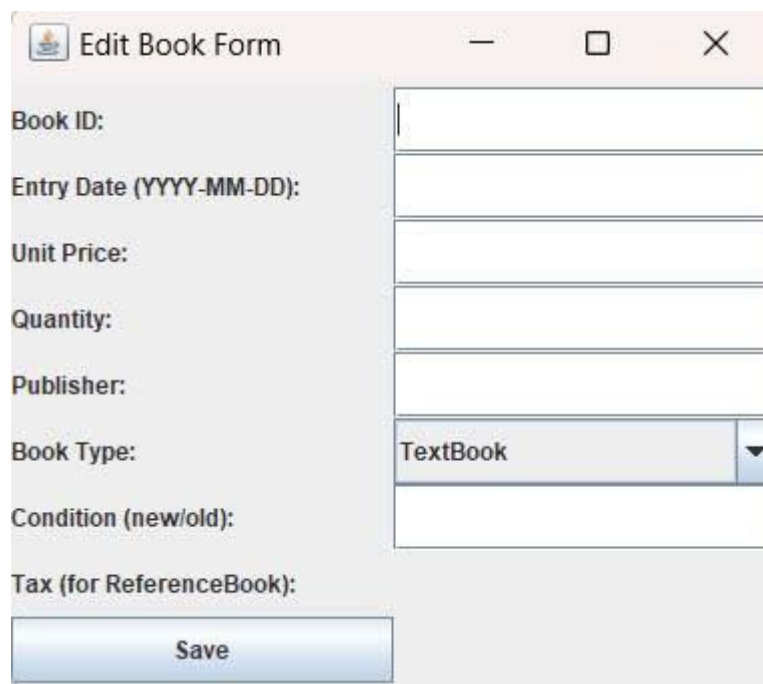
### 2.2. Chức năng thêm sách

The screenshot shows a window titled "Add Book Form" with standard Windows window controls (minimize, maximize, close). The form contains the following fields and controls:

- Book ID:** A text input field.
- Entry Date (YYYY-MM-DD):** A date input field.
- Unit Price:** A text input field.
- Quantity:** A text input field.
- Publisher:** A text input field.
- Book Type:** A dropdown menu currently displaying "TextBook".
- Condition (new/old):** A text input field.
- Tax (for ReferenceBook):** A text input field.
- Submit:** A blue button at the bottom of the form.

Hình 1.1 Chức năng thêm sách

### 2.3. Chức năng sửa sách



Edit Book Form

Book ID:

Entry Date (YYYY-MM-DD):

Unit Price:

Quantity:

Publisher:

Book Type:

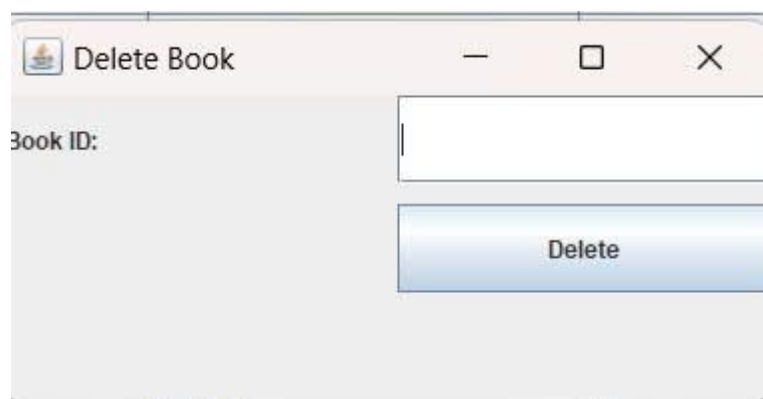
Condition (new/old):

Tax (for ReferenceBook):

Save

Hình 1.1 Chức năng sửa sách

### 2.4. Chức năng xóa sách




Delete Book

Book ID:

Delete

Hình 1.1 Chức năng xóa

### 2.5. Chức năng tìm kiếm



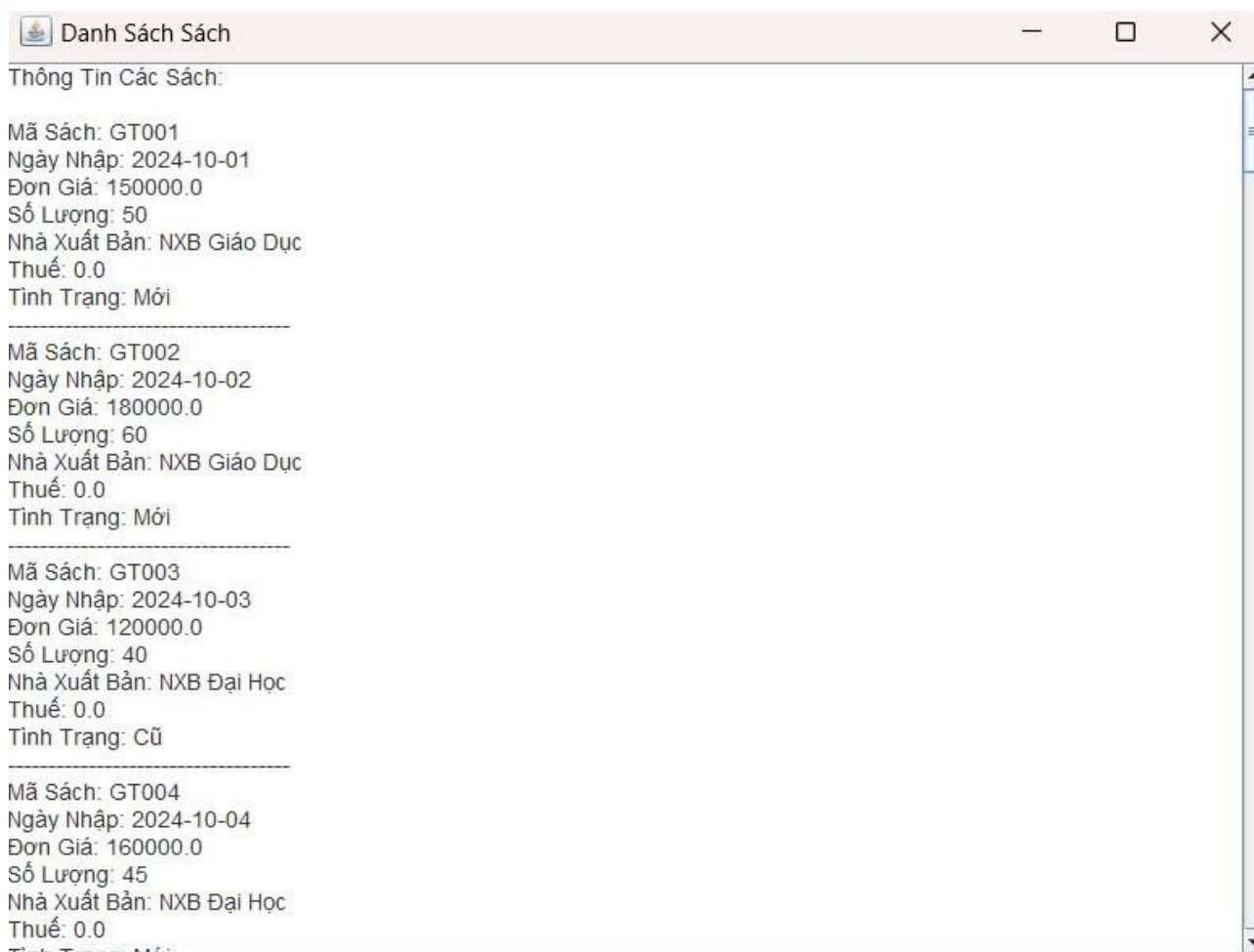
Search Book

Nhập id cần tìm:

Search

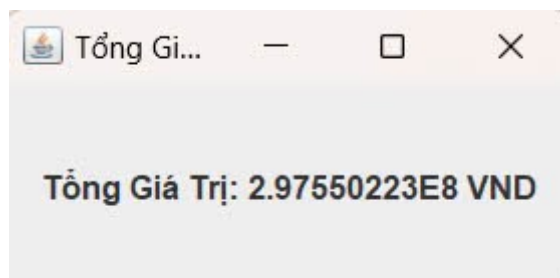
## Hình 1.1 Chức năng tìm kiếm

### 2.6. Chức năng in



Hình 1.1 Chức năng in

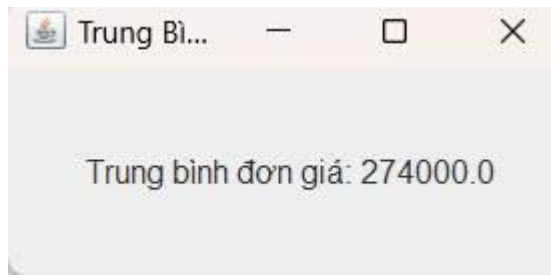
### 2.7. Chức năng tính tổng



Hình 1.1 Chức năng tính tổng

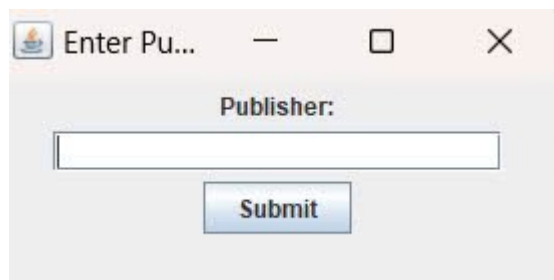


### ***2.8. Chức năng tính trung bình cộng***



**Hình 1.1 Chức năng tính trung bình cộng đơn giá sách tham khảo**

### ***2.9. Chức năng xuất sách theo nhà xuất bản***



**Hình 1.1 Chức năng xuất**

## HƯỚNG PHÁT TRIỂN

-Hệ thống báo cáo chi tiết: Xây dựng chức năng thống kê theo thời gian (theo ngày, tháng, năm), nhà xuất bản, hoặc loại sách. Ví dụ, báo cáo tổng số lượng và giá trị sách nhập trong tháng, hoặc phân tích doanh thu từng loại sách.

-Quản lý kho sách nâng cao: Thêm chức năng kiểm kê kho, tự động cảnh báo khi số lượng sách xuống thấp hoặc sắp hết. Điều này giúp quản lý việc nhập thêm sách kịp thời.

-Tìm kiếm và lọc sách đa dạng: Cải tiến tìm kiếm bằng cách hỗ trợ các tiêu chí phức hợp như ngày nhập, đơn giá, tình trạng sách, và nhà xuất bản. Đồng thời, tích hợp chức năng tìm kiếm gần đúng để hỗ trợ người dùng.

-Hệ thống phân quyền: Bổ sung chức năng đăng nhập và phân quyền cho người quản lý và nhân viên. Người quản lý có thể xem toàn bộ dữ liệu, trong khi nhân viên chỉ có quyền thêm hoặc sửa thông tin.

-Kết nối dữ liệu: Tích hợp hệ thống với cơ sở dữ liệu trực tuyến, cho phép lưu trữ và truy cập dữ liệu từ nhiều thiết bị.

Những hướng phát triển này không chỉ giúp hệ thống trở nên chuyên nghiệp hơn mà còn đảm bảo đáp ứng nhu cầu thực tế và mở rộng phạm vi ứng dụng.

## TÀI LIỆU THAM KHẢO

- Data. (2023). *[Tài liệu học SQL] Phần 1 – SQL là gì? Tổng quan về SQL cho người mới*. Retrieved from [https://datapot.vn/tai-lieu-tu-hoc-sql-tieng-viet-sieu-co-ban-chapter-1/?srsltid=AfmBOoqaEih8Oxb2\\_RwotUBSA77Q2psreO35JXnvKcxUslNu-fQQPaad](https://datapot.vn/tai-lieu-tu-hoc-sql-tieng-viet-sieu-co-ban-chapter-1/?srsltid=AfmBOoqaEih8Oxb2_RwotUBSA77Q2psreO35JXnvKcxUslNu-fQQPaad)
- GeeksforGeeks. (2024, 12 6). Retrieved from <https://www.geeksforgeeks.org/java-multithreading-tutorial/>
- misa. (2022). *Phân quyền trên Hệ thống*. Retrieved from <https://helpamis.misa.vn/kb/phan-quyen-tren-he-thong/>
- Trần, V. (2022). *Clean Architecture là gì - Ưu nhược và cách dùng hợp lý*. Retrieved from <https://200lab.io/blog/clean-architecture-uu-nhuoc-va-cach-dung-hop-ly/?srsltid=AfmBOopi9om1rui1YIPy1bCQ2nSsbuecRdNW8iqS64DN7jMyvGZ5EpXq>
- Trình, T. C. (2020). *Ứng Dụng Quản Lý Sách*. Retrieved from <https://www.studocu.com/vn/document/truong-dai-hoc-thu-dau-mot/cong-nghe-thong-tin/phat-trien-ung-dung-quan-li-sach/43235183>
- TypeScript. (2024, 11 22). *Tài liệu Hướng dẫn TypeScript*. Retrieved from <https://www.typescriptlang.org/docs/>